THESE DE DOCTORAT

presentée par

Laurence Emilie UM

Discipline: Informatique et Mathématiques Appliquées

Spécialité: Codes Correcteurs d'erreurs et Sécurité de l'Information

A CONTRIBUTION TO THE THEORY OF CONVOLUTIONAL CODES FROM SYSTEMS THEORY POINT OF VIEW

Laboratoire de Mathématiques Appliquées, Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat. Tel : + 212 (0) 37 77 18 34/35/38, Fax : + 212 (0) 37 77 42 61

Facultat de Matemàtiques y Estadística, Universitat Politècnica de Catalunya. BarcelonaTech, Campus Diagonal Sud, Edificio U. C. Pau Gargallo, 5 08028 Barcelona.

Tel: +34 93 401 58 80, Fax: +34 93 401 58 81

Acknowledgement

The work presented in this thesis were realized partly within the laboratory of mathematical and computing sciences in the Faculty of Sciences of Rabat, at the University Mohammed V-Rabat. That laboratory welcomed me with open arms and helped me experiment a warm and convivial atmosphere, and I would like to thank all of my colleagues for that experience. Also, we did relevant research at the Universitat Politècnica de Catalunya, at the department of Mathematics I and I would like to wholly thank the University for all the technical support and constant support even during harsh periods.

Indeed, my work on codes has been the mainspring and the catalyst behind the research project, not to forget to mention the constant growing need for storing and handling of information (rise of usage of mobile phones, internet communication, cloud computing), as well as data protection in general. Holding a master's degree in cryptography and information security from the University of Mohammed V-Rabat, I worked a lot on coding theory, also for academic projects.

I would like to first of all thank my thesis supervisors, who supported and accompanied me during that research period, despite the difficulties encountered. I would really like to thank Pr. El Mamoun Souidi for accepting me as a doctorate and supporting me, I would like to deeply thank the Universitat Politècnica de Catalunya in Spain, in particular Pr. Maria Isabel García-Planas for her advice very much needed, the constant push, for the help material or intellectual, for her hospitality in Spain, presence at any time needed, on top of accepting to welcome me within their structure for the realization of the thesis in co-supervision.

I would like to thank my beloved family; in particular my amazing and beloved parents, who despite the distance have always been a constant, invaluable and continuous support of my work and progress throughout my life in general, and student life in particular, and without whom I absolutely would

not have been able to make it this far.

I would like to thank my friends and colleagues for the encouragement, understanding and moral support; as it happens, many warm thanks to Edouard.

List of Figures

Abstract

Information is such a valuable good of our time. Given that the transmission of information has always been subject to precision problems, knowing the obstacles existing between the transmitter and the receiver, eventual disruptions can happen anywhere in between, the physical means, channels involved with the exchange are never perfect and they are subject to errors that might result in loss of important data.

Given that error correcting codes are a key element in the transmission and storage of digital information, therefore an easier and better usage opens up to interesting opportunities in regulating transmission of information, which is the advantage that the linear systems theory brings definition of convolutional codes, along with the algebra material. It is the reason why in this thesis, we follow that perspective to study the possibility to redefine and improve properties of convolutional codes in terms of coding and decoding, with the help of the systems and control theory.

For that matter, in chapter 1, we recall notions on coding theory, more specifically, on linear codes, both block and convolutional, redefining the convolutional codes as submodules of $\mathbb{F}_q^n$ which is our main workspace. And we go through the prerequisites involved in the process of encoding and decoding, both for block and convolutional codes.

And in order to approach them with tools of the systems theory, in chapter 2, we give the equivalence of the generating matrix in the form of a realization (A, B, C, D) of an input-output system. Then, we studied the concatenation because it has been proved to improve the transmission. In this work, we consider two big families of concatenation: serial concatenation, and parallel concatenation and two other models of concatenation called systematic serial concatenation and parallel interleaver concatenation.

In chapter 3, we study control properties for each case. Nevertheless, we

focus on the property of "functional output controllability"in coding theory language is called "output-observability", and conditions to obtain it, particularly an easy iterative test is presented in order to discuss whether a code is output-observable. This test consists in calculating certain ranks of block matrices constructed from the matrices A, B, C, D. The output-observability property is very beneficial for the decoding as discussed in the next chapter.

Moreover, in chapter 4, we assess two methods for a complete decoding operating on an iterative fashion, then suggest conditions for a step by step decoding in a case of concatenation, in order to recover exactly each and every original sequence after operation of every implied code. Following this concept, we study the convolutional decoding in general, and in particular the one of concatenated models in serial, in parallel, in systematic serial and finally in interleaver parallel implementation.

In chapter 5, we suggest an application in steganography, in which we implement a steganographic scheme, inspired by the linear system representation of convolutional codes. Having the output-observability matrix being the backbone behind the construction of our decoding algorithms, coupled with the syndrome method, we formed some embedding/retrieval algorithms inspired by that construction. Those methods display the protection of communication within time-related transfer of information, with interesting possibilities and results.

Finally, a chapter summarizing all our achievements, within which the establishment of a new realization building algorithm, methods and algorithms to solve the decoding of convolutional codes. This application of linear systems within the convolutional codes theory showed a range of possibilities for us to explore, since as an additional application, we developed some new steganographic models, based on the representation of convolutional codes within the linear systems theory, and also a short list of possible future lines of work that we would like to continue studying in order to achieve new related goals.

Résumé

L'information est un bien de notre époque dont l'importance n'est plus à démontrer. Etant donné que la transmission de l'information a toujours été soumise à des problèmes de précisions, dûs aux obstacles existant entre le transmetteur and le recepteur, d'éventuelles pertubations peuvent arriver n'importe où, entre les canaux physiques, faisant partie du processus d'échange qui n'est jamais parfait et ils peuvent toujours être affectés par des erreurs créant d'importantes pertes d'information. Les codes correcteurs d'erreurs sont un élément clé dans la transmission et la conservation de l'information numérique.

Etant donné que les codes correcteurs d'erreurs sont un élement clé dans la transmission et la conservation de l'information digitale, ainsi un meilleur et plus simple usage ouvre des opportunités plus intéressantes dans la régulation de la transmission de l'information, qui est l'avantage que la définition des codes convolutifs suivant la théorie des systèmes linéaires apporte, avec le matériel de l'algèbre linéaire. C'est pour cette raison que dans cette thèse, nous suivons cette perspective pour étudier la perspective d'étudier la possibilité de redéfinir et d'améliorer les propriétés des codes convolutifs en terme de codage et de décodage, grâce aux outils de la théorie des systèmes et de contrôle.

A cet effet, dans le chapitre 1, nous rappelons des notions sur la théorie des codes linéaires, les codes en bloc ainsi que les codes convolutifs, redéfinissant les codes convolutifs comme des sous-modules de $\mathbb{F}_q^n$ qui est notre principal espace de travail. Et c'est ainsi que nous invoquons tous les pré-réquis nécessaires pour le processus de codage et de décodage, pour ce qui est des codes en bloc, et des codes convolutifs.

Et dans le but d'approcher ces derniers grâce aux outils de la théorie des systèmes, dans le chapitre 2, nous donnons l'équivalence de la matrice génératrice sous la forme d'une réalisation (A, B, C, D) d'un un système input-output. Ensuite, nous étudions la concatenation parce qu'elle a été prouvée d'améliorer la transmission. Pour cette partie, nous considérons deux grandes

familles de concaténation: concaténation en série et en parallèle, ainsi que deux autres modèles de concaténation appelés: concaténation systématique en série et concaténation en parallèle avec interleaver.

Dans le chapitre 3, nous étudions les propriétés de contrôle pour chacun des cas. Néanmoins, nous nous concentrons sur la propriété de "functional output controllability"que dans le langage de théorie est appelé "output-observability", et sur les conditions pour l'obtenir, en particulier un test itératif relativement facile a été présenté en vue de discerner les codes output-observables de ceux qui ne le sont pas. Ce test permet de calculer certains rangs de blocs de matrices construits à partir des matrices A, B, C, D. La propriété d'output-observabilité est très bénéfique pour le décodage comme explicité dans le prochain chapitre.

De plus, dans le chapitre 4, nous évaluons deux méthodes pour un décodage complet opérant de manière itérative, ensuite suggérons des conditions pour un décodage étape par étape dans un cas de concaténation, en vue de recuperer exactement chacune des séquences d'origine après opération de chacun des codes impliqués. Suivant ce concept, nous étudions le décodage convolutif en géneral et en particulier celui des modèles de concaténation en série, en parallèle, en série systématique et finalement en parallèle avec interleaver.

Dans le chapitre 5, nous suggérons une application en stéganographie, dans laquelle nous implémentons un schéma steganographique, inspiré par la représentation en termes de systèmes linéaires des codes convolutifs. Ayant la matrice d'output-observabilité étant la matrice de réference pour la construction de nos algorithmes de décodage, couplée avec la méthode du syndrome, nous avons proposé quelques algorithmes d'encapsulation et de recouvrement inspirés par cette construction. Ces méthodes montrent la protection de la communication lors des transferts d'information dépendant du temps, avec d'intéressantes possibilités ainsi que des résultats encourageants.

Finalement, un chapitre résumant tout ce que nous avons accompli, en l'occurence la mise sur pied d'un nouvel algorithme pour écrire une réalisation, méthodes et algorithmes pour résoudre le décodage des codes convolutifs. Cette application des systèmes linéaires sur la théorie des codes convolutifs montre un ensemble de possibilités pour nous à explorer, puisque nous avons développé une application de plus , nous avons developpé quelques modèles stéganographiques, basés sur la représentation des codes convolutifs grâce à la théorie des systèmes linéaires, et une courte liste des futurs possibles axes de travail sur des aspects que nous souhaiterions étudier pour parachever nos

buts traitant de problématiques similaires.

Resumen

La información es un valioso bien de nuestro tiempo. Dado que la transmisión de la información siempre ha estado sujeta a problemas de precisión, conociendo los obstáculos existentes entre el transmisor y el receptor, las interrupciones eventuales pueden ocurrir en cualquier lugar en el medio, el medio físico, canal involucrado con el cambio nunca es perfecto y está sujeto a errores que podrían dar como resultado una pérdida de datos importantes.

Dado que los códigos correctores de errores son un elemento clave en la transmisión y almacenamiento de información digital, por eso un más fácil y mejor uso abre interesantes oportunidades en la regulación de la transmisión de la información, el cual es una ventaja que ofrece la teoría de sistemas lineales y el álgebra lineal a la definición de los códigos de convolución . Esta es la razón por la que en esta tesis, seguimos esa perspectiva para estudiar la posibilidad de redefinir y mejorar las propiedades de los códigos de convolución en base a la codificación y descodificación, con la ayuda de los sistemas y la teoría de control.

En este sentido, en el capítulo 1, recordamos nociones sobre la teoría de códigos, más específicamente, sobre los códigos lineales, tanto de bloques como de convolución, se redefinen los códigos convolucionales como submódulos de $\mathbb{F}_q^n$ que es nuestro espacio principal de trabajo. Y damos un repaso a los requisitos previos necesarios en el proceso de codificación y descodificación, tanto para los códigos de bloque como los códigos convolucionales.

Y con el fin de aproximarnos a los códigos convolucionales con las herramientas de la teoría de sistemas, en el capítulo 2, damos la equivalencia de la matriz generatriz en función de una realización (A, B, C, D) de un sistema de entrada-salida. A continuación, se estudia la concatenación porque es conocido que mejora la transmisión. En este trabajo, se consideran dos grandes familias de concatenación: la concatenación en serie, y la concatenación en paralelo así como otros dos modelos de concatenación llamados concatenación en serie

sistemática y la concatenación en paralelo con intercalador.

En el capítulo 3, estudiamos propiedades de control para cada caso. Sin embargo, nos hemos centrado en la propiedad de "funcional output-controlabilidad" que en lenguaje de teoría de códigos es conocido como "output-observabilidad", y en obtener condiciones que aseguren dicha condición, en particular se presenta un fácil test iterativo, que permite discutir cuando un código de convolución es output-observable. Este test consiste en calcular los rangos de ciertas matrices por bloques construidas a partir de las matrices A, B, C, D. La propiedad de output-observabilidad es muy útil para la descodificación que se estudia en el próximo capítulo.

Por otra parte, en el capítulo 4, se presentan dos métodos para una completa descodificación operando de forma iterativa, a partir de ahí, se sugieren condiciones para paso a paso descodificar la concatenación, a fin de recuperar exactamente todos y cada uno de los códigos implicados en la operación. Siguiendo esta idea, se estudia la descodificación de los códigos convolucionales en general, y en particular la de los modelos concatenados en serie, en paralelo, en serie sistemática y finalmente la concatenación en paralelo con intercalador.

En el capítulo 5, se presenta una aplicación a la esteganografía, en el que se implementa un esquema esteganográfico, inspirado en la representación del sistema lineal de códigos convolucionales. La matriz de output-observabilidad es la columna vertebral que está detrás de la construcción de nuestros algoritmos de descodificación que junto con el método de síndrome, formamos algunos algoritmos Inclusión/recuperación inspirados en esa construcción. Estos métodos muestran la protección de la comunicación dentro de la transferencia relacionada con el tiempo que dura la información, con interesantes posibilidades y resultados.

Por último, un capítulo que resume todos nuestros logros, en este caso el desarrollo de un nuevo algoritmo para escribir una realización, los métodos y algoritmos para resolver la descodificación de códigos convolucionales. Esta aplicación a los códigos convolucionales de la teoría de sistemas lineales muestra un abanico de oportunidades para explorar, ya que como una aplicación adicional, hemos desarrollado algunos nuevos modelos esteganográficos, basados en la representación de los códigos convolucionales usando la teoría de sistemas lineal, y una corta lista de posibles futuras líneas de trabajo en los aspectos que nos gustaría seguir estudiando para alcanzar nuevas metas relacionadas seguir estudiando para alcanzar nuevas metas relacionadas con este tema.

Contents

Introduction

The transmission of information has always been subject to precision problems, given the obstacles existing between the transmitting and the receiver, especially within the scope when both are subject to distance constraints [92]. Then, the fundamentals to ensure an efficient transfer such as availability (efficient use of the network, the time delays) or traceability at the reception (to ensure non-rejection) when getting to the recipient are to be met. The need to ensure quality transfer goes back to the delivery of messages by illustrations carved in stone, until invention of the writing, where more modern methods have been requested.

Indeed, my work on codes has been the mainspring and the catalyst behind the research project, not to forget to mention the constant growing need for storing and handling of information (rise of usage of mobile phones, internet communication, cloud computing), as well as data protection in general. Holding a master's degree in cryptography and information security from the University of Mohammed V-Rabat, I worked a lot on coding theory, also for academic projects. Given the already known link with the systems theory, the idea of combination of those two theories in order to consider convolutional codes under a different angle different from the classical one were suggested.

The information theory also known as Shannon's information theory, was officially born in 1948 [75]. This theory helps treat raw information in a set of sections and/or partitions, in an acceptable format by the transmission channel, in order to ensure a coherent and adequate distribution, corresponding to the utilization needs (messages, different sort of data, etc...) see [37]. This last is represented by many other theories, such as the coding one (see [93]).

In 1948, Claude Shannon landed the first stone of what he called a "mathematical theory of information". It is said that its theory was born from thoughts on the language (english as it happens): Shannon was trying to mask a variable proportion of text that he was reading, and to piece them

together from the visible part. Because those hidden words were redundant, they wouldn't bring anything more to the meaning of the message. If he took out too much, he could reconstruct the message with certainty. Then Shannon put in place a theory that would be able to compute the quantity of information in any type of message which comes down to determining the rate of redundancy (for more, see [18]).

The coding theory which is part of the big picture, rather focuses on efficiency of messages transmission, in order to guarantee the integrity at the moment of reception by the addressee (For more information, see [9], [11], [49], [86]). Then, it actually tackles the storage question, and also the sending and reception via a channel, in the aim to minimize the disturbances generally encountered when transferring data.

At the origin, coding theory has had mainly the fundamental dedication on information theory. In fact coding theory had arisen from the need for better communication and better computer data storage (see [42]). Getting to grips with the problem generally goes by tackling at first the coding, and then the decoding or vice-versa. The coding theory is itself a very large and deep subject (see [39], [57], [65], [40]). There are various ways to implement the coding theory, such as network coding [8], or LRC coding for instance, as well as for the decoding as list decoding for instance [32], [80]. Usually, the decoding is one of the most challenging problem to solve; as we can imagine, for convolutional sequences, they are semi-infinite, which means that the decoding complexity is one not an ignorable issue. The most known one is the Viterbi algorithm, based on maximum likelihood. Aside from that, there is also the Massey's threshold decoding as well as sequential and feedback algorithms [54], [41], [61]. Also, some iterative decoding have been introduced in the field, prior to the low density parity check codes, and the general graphs based methods [82], [85] (for much more information, see [67]).

In coding theory, a linear code is an error-correcting code for which any linear combination of codewords is also a codeword. It is the guarantee of confidence for non alteration in numerical transmission via a channel. Following the linear algebra rules, a linear code to be defined needs three sets, the first being the alphabet, the set of all the symbols used for messages; the second being a set of messages to be sent, which will be designated by M, the last being a set of messages to be received, which will be designated by R. Indeed, we use a lot of algebra notions. The process of messages transmission from M to R is then insured by a linear encoding matrix G, and the decoding by another function ψ. If we look specifically at the encoding process, which is supposed

to provide the non alterability of sent messages, the idea is to add redundance to them. On the other side, the decoding process verifies the veracity of the messages received.

During the first half of the twentieth century, linear systems were analyzed using frequency domain (e.g., Laplace and Z-transform) based approaches in an effort to deal with issues such as noise and bandwidth issues in communication systems. While they provided a great deal of intuition and were sufficient to establish some fundamental results, frequency domain approaches presented various drawbacks when control scientists started studying more complicated systems (containing multiple inputs and outputs, nonlinearities, noise, and so forth). Starting in the 1950s (around the time of the space race), control engineers and scientists started turning to state-space models of control systems in order to address some of these issues. These time-domain approaches are able to effectively represent concepts such as the internal state of the system, and also present a method to introduce optimality conditions into the controller design procedure, [81].

Concretely convolutional codes are extensively used in many wireless transmissions systems such as transmitting information in deep space with remarkable clarity. These codes are oftentimes implemented in concatenation with a hard-decision code, particularly Reed Solomon [59]. Before turbo codes [2], such constructions were the most efficient, coming closest to the Shannon limit. Recently, as we are working on convolutional codes over linear systems, some interesting proposals have resurfaced, knowing the fact that we are in an era of information. Indeed, information is so much powerful and present in our life than ever, and of course dealing with instantaneous and important flow of data requires means of supervising the exchanges all over the internet(video streaming for instance). Convolutional codes then find their spot and importance, as well as decoding.

The initial work on connection between the coding theory and the systems theory derives from a workshop initiated by Paul Fuhrmann at the "Institute for Mathematics and its Applications"(IMA) in August 1999 [66]. Within that meeting, Paul Fuhrmann suggested to the participants to establish diverse approaches and eventually unusual of convolutional codes from their own perspectives. Which generated a variety of points of view, for example the dynamic systems point of view resulting from the analogy existing between an input-output system and the "LFSR"(Linear Feedback Shift Register) which is the way used for the storage and encoding of information for the convolutional codes. The work realized in that area go from J. Rosenthal, R. Smarandache

on [66],[35], to J-J. Climent [12]; those inspired and brought us, and they are also the ones we based our contribution on; we also tried to bring our personal contribution and continue on their results.

From our perception, handling the convolutional codes using the algebraic approach has shown tremendous advantages, considering not only an easier manner to handle construction, but computationally speaking as well. Our own contributions are visible with coding, decoding and an additional application with steganography. When it comes to coding, it feels very convenient to be able to move from one side to another, algebraic or vector-space model corresponding to the context of the needs of the user, there is much more flexibility. Specially for decoding, some decoding algorithms have been suggested two years ago, such as the decoding over the erasure channel [72], as we are also trying to propose decoding procedures using the algebraic technique over the Hamming metric.

This thesis has been built around five chapters. We started by giving the preliminaries that are involved with our work, which is talking about the linear block codes and convolutional codes, and also introduced the notions involved in that theory, such as the distance, the length, the dimension, the weight and the ratio. Specifically, for the convolutional codes, those notions are defined and interpreted a slightly different way, which is as in a time-depending transaction. In fact, we illustrate the fact that convolutional codes are implemented with LFSR, that are registers characterized by their memory very helpful for saving in that situation of time dependance. We present some concrete examples to clearly show the encoding and the decoding in both cases. The second chapter displays the systems theory, as it starts by the input-state-output representation and the quadruple of matrices (A, B, C, D) involved in that representation. It also shows the theory of linear systems, and we give the representation of convolutional codes as linear systems as well. The third chapter assembles the control theory, in where start our contributions; in this one, we assess properties of the system theory with conditions in order to reach those, properties such as controllability and observability, and a new one introduced called output-observability; it represents the possibility of an internal state, to be only defined by a finite set of outputs, for a finite number of steps. Indeed, we use this concept in order to assess the minimality of a realization, which translates into minimality of a convolutional encoder. The fourth chapter is about algorithms for the decoding of the several constructions of convolutional codes we worked on. On the fifth chapter, we introduced steganography based on convolutional codes, inspired by steganography based

on coding theory in general. And on the last chapter, we have a summary of all of our achievements with a list of other aspects we would like to continue working in the future.

Chapter 1

Linear block and convolutional codes

The information theory deals with the means to quantify the encoding of information and its transmission in noisy channels. In this chapter, We recall notions on coding theory. Most specifically, We focus on notions on linear codes, both block and convolutional; we go through the prerequisites involved in the process of encoding and decoding, both for block and convolutional codes.

1.1 Linear block codes

In coding theory, block codes refer to the large and important family of error-correcting codes that encode data in blocks. (For more information, see [73]).

We consider a finite set of symbols $\mathbb{F}_q$, called alphabet, with q elements. The information to be processed and the codewords will be expressed with symbols from this alphabet. The set $\mathbb{F}_q$ is structured as a finite field (in particular the size q of the alphabet is a power of a prime number).

If confusion is not possible, we will denote the finite field $\mathbb{F}_q$ by $\mathbb{F}$.

Definition 1.1.1. A linear block code of length n and dimension k is a k-dimensional subspace C of the vector space $\mathbb{F}_q^n$.

The block code is referred as $C(n, k)$. The length of the code fixes the length

of the data streams sent through the channel, and the dimension measures the amount of information, without redundancy, that each of these streams has.

The block code can also be characterised by a size called ratio defined by:

Definition 1.1.2. A ratio of a linear block code is given by the expression: k/n, where n is the length of the code, and k is its dimension.

Encoding is described by means of an injective linear map called encoding map,

$$g : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$$

with image space $g(\mathbb{F}_q^k) = C$.

The set $\mathbb{F}_q^k$ is called the set of information words, and each element of the code $C = \{v \times G, \forall\, v \in \mathbb{F}_q^k\}$ is a codeword, where $G \in M_{k\times n}(\mathbb{F}_q)$ is the matrix associated to the linear map, called generator matrix.

Example 1.1.1. Let us consider a binary linear block code $C(6, 2)$, defined by the matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Since we have a binary code, our finite field is $\mathbb{Z}/2\mathbb{Z}$; and we have $n = 6$ and $k = 2$;

Taking into account that $\mathbb{F}_2^2 = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Then, the code is the linear subspace of dimension 2:

$$C = \{000000, 101101, 000111, 101010\}.$$

When defining error correcting codes, the notion of distance is very important. It is the decisive factor when it comes to the decoding, since the verification of a good data transmission depends on the distance between the received word, and those in the codewords space. This notion is defined by Richard Hamming in one of the first articles defining the encoding theory ([33]).

Definition 1.1.3. Let $x \in \mathbb{F}^n$ be a vector, we define the Hamming weight of x as the number of nonzero components of the n-vector x. We will denote the Hamming weight of x by $w(x)$.

Definition 1.1.4. If x_1, x_2 are two vectors in $\mathbb{F}^n$, we define the Hamming distance dist (x_1, x_2) through the formula dist $(x_1, x_2) = w(x_1 - x_2)$.

As before, the Hamming distance satisfies all axioms of the Euclidean metric defined in $\mathbb{R}^n$.

Definition 1.1.5. The minimal distance is defined by either the smallest distance between two words, or the smallest non zero Hamming weight of all codewords.

Example 1.1.2. Let us compute the minimal distance of some specific codes.

In $\mathbb{F}_2$, let us have the code $C_1(4, 2)$ given by the generator matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

The family of codewords generated by encoding is

$C_1 = \{0000, 1011, 0101, 1110\}$.

For this code, the minimal distance is 2, the weight of the word 0101.

In $\mathbb{F}_3$, we consider the code $C_2(4, 2)$ given by the following generating matrix

$$\begin{pmatrix} 0 & 1 & 2 & 1 \\ 2 & 0 & 1 & 0 \end{pmatrix}.$$

The codewords belonging to this code are:

$C_2 = \{0000, 0121, 2010, 2101, 0212, 1020, 1202, 2222, 1111, 1202\}$.

The minimal distance is 2, the weight of both 2010 and 1020.

In coding theory the usual criterion for representation of generator matrices is the horizontal which means a (k, n) representation and encoding from the left. From the next section and the rest of the chapters, we will keep the vertical representation because it is usual criterion in linear systems theory.

1.2 Convolutional codes

Convolutional code is a type of error-correcting code in which each k-bit information symbol (each k-bit string) to be encoded is transformed into an n-bit symbol, where k/n is the code rate and the transformation is a function of the last information symbols contained in the memory of the physical encoder.

Convolutional codes are used extensively in numerous applications in order to achieve reliable data transfer, including digital video, radio, mobile communication, and satellite communication (See [3]). These codes are often implemented in concatenation with a hard-decision code, particularly Reed Solomon code. Prior to turbo codes, such constructions were the most efficient, coming closest to the Shannon limit. Convolutional codes were considered for the first time by Elias in [19].

Convolutional codes are an improvement of block codes. They were implemented in order to allow error correcting codes, to encode a longer sequence of block words, at the same time, and have a better efficiency for error correction, as well as encoding and decoding. The principle was, for a sequence of blocks $m_1, m_2, ..., m_s \in \mathbb{F}^k$, to be encoded, to introduce a polynomial vector, that would reunite them all, with $m(z) = \sum_{i=0}^{s} m_i z_i \in \mathbb{F}^k[z]$, and transmit them with a polynomial encoder [61]; to do so, we define a matrix G, entries in $\mathbb{F}[z]$, we get $c(z) = G(z)m(z) \in \mathbb{F}^n[z]$, ([66]).

Definition 1.2.1. A convolutional code of length n and dimension k is a k-rank submodule of $\mathbb{F}_q^n[z]$.

Remember that

Definition 1.2.2. Let A be a ring. An A-module M is an additive abelian group M equipped with an action:

$$\begin{array}{rcl} A \times M & \to & M \\ (a, m) & \mapsto & am \end{array} \tag{1.1}$$

satisfying the conditions:

1. $a(m_1 + m_2) = am_1 + am_2$ for each $a \in A$ and $m_1, m_2 \in M$;
2. $(a_1 + a_2)m = a_1 m + a_2 m$ for each $a_1, a_2 \in A$ and $m \in M$;

3. $(a_1a_2)m = a_1(a_2m)$ for each $a_1, a_2 \in A$ and $m \in M$.

Definition 1.2.3. Let A be a ring. Given an A-module M, an A-submodule N $\subset$ M is an additive subgroup such that, for each $a \in A$ and $n \in N$, we have $an \in N$.

The word "convolutional" is used because the output sequences can be regarded as the convolution of the input sequences with the sequences in the encoder (see [20]).

Corollary 1.2.1 ([70])**.** *Let C be a convolutional code. Then, there exists an injective morphism of modules*

$$\begin{array}{rl} \psi : \mathbb{F}^k[z] & \mapsto \mathbb{F}^n[z] \\ u(z) & \mapsto v(z). \end{array} \tag{1.2}$$

Equivalently, there exists a polynomial matrix $G(z)$ (called encoder) of order $n \times k$ and having maximal rank such that

$$\mathcal{C} = \{v(z) \mid \exists u(z) \in \mathbb{F}^k[z], v(z) = G(z)u(z)\}. \tag{1.3}$$

The rate k/n is known as the ratio of a convolutional code. We denote by ν_i the maximum of all degrees of each of the polynomials of each line, we define the complexity of the encoder as $\overline{\delta} = \sum_{i=1}^{n} \nu_i$, and finally we define the complexity of a convolutional code $\delta(\mathcal{C})$ as the maximum of all degrees of the largest minors of $G(z)$ that we will write simply by δ if no confusion is possible [41].

The representation of a code among relatively different representations by means of a polynomial matrix is not unique, but we have the following proposition.

Proposition 1.2.1 ([70])**.** *Two $n \times k$ rational encoders $G_1(z)$, and $G_2(z)$ define the same convolutional code, if and only if there exists a $k \times k$ unimodular matrix $U(z)$ such that $G_1(z)U(z) = G_2(z)$.*

Remember that

Definition 1.2.4. A polynomial matrix $P(z) \in \mathbb{F}[z]$ is unimodular if there exists another matrix $Q(z)$ such that $P(z)Q(z) = I$.

Equivalently, a polynomial matrix $P(z) \in \mathbb{F}[z]$ is unimodular if and only if $\det P(z)$ is a non-zero element of the field $\mathbb{F}$. Notice that the "inverse" $Q(z)$ is also an invertible polynomial matrix.

After a suitable permutation of the rows, we can assume that the generator matrix $G(z)$ is in the form

$$G(z) = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix} \tag{1.4}$$

with right coprime polynomial factors (block of polynomials) $P(z) \in \mathbb{F}_{(n-k)\times k}$ and $Q(z) \in \mathbb{F}_{k\times k}$, respectively.

It is possible to consider the equivalent rational encoder where $Q(z) \neq 0$

$$\begin{pmatrix} P(z) \\ Q(z) \end{pmatrix} Q^{-1}(z) = \begin{pmatrix} P(z)Q^{-1}(z) \\ I \end{pmatrix}. \tag{1.5}$$

Such operation can be implemented through shift registers. Those Linear Feedback Shift Registers (LFSR) can be characterized by their memory, which represents the number of delays realized by each one, while encoding every bit of information.

Depending on the type of the shift registers, convolutional codes can be qualified in accordance with 3 properties: systematic recursive, systematic nonrecursive and nonsystematic nonrecursive [51].

The encoder is said to be systematic if the output bits are reproduced transparently in the transmitted stream; and the opposite when none of the outputs is transparently similar. The recursivity, as the name suggests comes when there is feedback into the input.

Indeed, it is important to denote that encoding a message, is pretty similar to the block code encoding, with the generator matrix; the small difference will be specific to the type of code we are dealing with, especially because of the interlacement. However, it is much easier than the decoding, whose algorithms can get quickly complex, and that could be seen as the weakness of convolutional codes.

Example 1.2.1. The convolutional code C corresponding to Figure 1.1 is given by

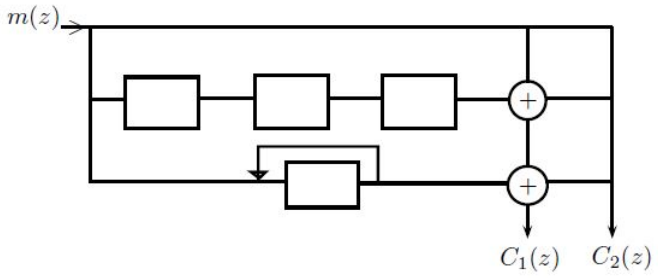

Figure 1.1: A representation of a recursive encoder

$$C_1(z) = z^3 m(z) + \frac{z}{1+z} m(z)$$
$$C_2(z) = m(z).$$

The encoding matrix corresponding to our encoder is given by:

$$G = \begin{pmatrix} z^3 + \dfrac{z}{1+z} \\ 1 \end{pmatrix}.$$

We can have an equivalent polynomial matrix, with the product $G(1+z)$ that is equivalent to a proper rational matrix.

$$G(1+z) = \begin{pmatrix} z + z^3 + z^4 \\ 1+z \end{pmatrix} \sim \begin{pmatrix} 1+z \\ z + z^3 + z^4 \end{pmatrix} \sim \begin{pmatrix} \dfrac{1+z}{z + z^3 + z^4} \\ 1 \end{pmatrix}.$$

Definition 1.2.5. The state diagram is a diagram that shows the different possible states that can be taken by the encoder, depending on the entering bit, and how to move from one another.

(An example is showed in Figure 1.4)

Definition 1.2.6. The state table is a table whose entries are filled in with bits contained in the shift registers either at present (two last entries of each row) or next time(two first entries of each row).

(An example is showed in Figure 1.2)

Many other different definitions of convolutional codes can be found in the literature, as suggested by [66]. In his work, J. Rosenthal details all of the different definitions over "the linear algebra, the symbolic dynamics, the linear time-invariant behavior, and the first order representation definition", as well as their relevance and shows the equivalence between them. The one to pick depends on the goal we are pursuing. For our work, especially, our aim is to benefit from the linear systems theory, to define some algebraic properties, and always improve convolutional codes, by privileging the concatenation, and the decoding algorithms as well. One of the highly used is the Viterbi algorithm. Another algorithm is given by Massey for decoding BCH codes by a shift register approach (For more information, see [54]). Therefore, We will be focusing, for the rest of our work, on a linear systems theory definition of convolutional codes. One of the best representation We found corresponding to illustrate this connection will be the first-order representation.

As for the linear block codes, let us consider a Galois finite field, our basic finite field. Then, $\mathbb{F}^n$ is our arrival space.

Theorem 1.2.1. *Let $C \subset \mathbb{F}^n$ be a k/n convolutional code, of complexity of convolutional code δ. Then, there exist matrices K, L of sizes $(\delta + n - k) \times \delta$, and a matrix M, of size $(\delta + n - k) \times n$, with entries in $\mathbb{F}$, such that the convolutional code C is defined by*

$$C = \{u(z) \in \mathbb{F}^n[z] | \exists x(z) \in \mathbb{F}^\delta[z] : zKx(z) + Lx(z) + Mu(z) = 0\}.$$

Example 1.2.2. We consider the following specific model of encoding:

Let $G(z) = \begin{pmatrix} z^2 & z+1 \\ z^2+z+1 & 1 \\ 1 & z \end{pmatrix}$ be the generator matrix of a code with $\frac{k}{n} = \frac{2}{3}$, $[\nu_i] = [\nu_1, \nu_2] = [2, 1]$ where ν_1 and ν_2 represent the highest degrees of respectively the first and second columns of $G(z)$, then, the complexity of convolution code is $\delta = \nu_1 + \nu_2 = 3$.

Consider the matrix

$$X(z) = \begin{pmatrix} 1 & 0 \\ z & 0 \\ 0 & 1 \end{pmatrix} \in M_{\delta \times k}(\mathbb{F})$$

such has maximal rank, $\operatorname{rank} X(z) = k$, then for each $f(z) \in \mathbb{F}^k[z]$ there exists $v \in \mathbb{F}^\delta$, such that $vX(z) = f(z)$

Let $\begin{pmatrix} 0&1&0&0&0\\0&0&1&0&0\\0&0&0&0&1\\1&0&0&0&0\\0&1&0&0&0\\0&0&0&1&0\\0&0&1&1&1\\1&1&1&1&0\\1&0&0&0&1 \end{pmatrix}$ be the scalar matrix of $Q(z) = \begin{pmatrix} zX(z)\\X(z)\\G(z)\end{pmatrix}$.

We have

$$\begin{aligned}\Lambda : \mathbb{F}^{2\delta+n} &\mapsto \mathbb{F}^{\delta+k}\\ v &\mapsto vQ(z)\end{aligned}$$

We obtain $\operatorname{Ker}\Lambda$ by solving

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9\end{pmatrix}\begin{pmatrix} 0&1&0&0&0\\0&0&1&0&0\\0&0&0&0&1\\1&0&0&0&0\\0&1&0&0&0\\0&0&0&1&0\\0&0&1&1&1\\1&1&1&1&0\\1&0&0&0&1 \end{pmatrix} = 0$$

The subspace $\operatorname{Ker}\Lambda$ is the subspace generated by the row vectors

$$\begin{pmatrix} 1&0&0&0&1&0&0&0&0\\0&1&1&0&0&1&1&0&0\\0&1&0&1&1&1&0&1&0\\0&0&1&1&0&0&0&0&1\end{pmatrix}$$

Then,

$$K = \begin{pmatrix}1&0&0\\0&1&1\\0&1&0\\0&0&1\end{pmatrix} \in M_{(\delta+n-k)\times\delta}(\mathbb{F}), \quad L = \begin{pmatrix}0&1&0\\0&0&1\\1&1&1\\1&0&0\end{pmatrix} \in M_{(\delta+n-k)\times\delta}(\mathbb{F}),$$

$$M = \begin{pmatrix}0&0&0\\1&0&0\\0&1&0\\0&0&1\end{pmatrix} \in M_{(\delta+n-k)\times n}(\mathbb{F}).$$

Obviously, $\begin{pmatrix} zK & L & M \end{pmatrix} \begin{pmatrix} X \\ X \\ G \end{pmatrix} = zKX + LX + MG = 0.$

The definition of Hamming distance given in 1.1.4, can be extended to vector spaces in the following manner

Definition 1.2.7. The free distance represents the minimal distance between two codewords, at a specific time of the encoding.

The free distance can be assimilated to the minimal distance as interpreted with block codes when it comes to convolutional codes.

Example 1.2.3. In $\mathbb{F}_2$, let us consider the non-recursive and non-systematic convolutional code $\mathcal{C}$, of ratio $1/2$, whose matrix is defined below.
As mentioned earlier, we consider convolutional encoders as matrices with rational or polynomial entries. Considering that we operate at the right, the polynomial generator matrix is given by:

$$G(z) = \begin{pmatrix} 1 + z + z^2 \\ 1 + z^2 \end{pmatrix}.$$

In order to make it easier to compute the outputs, knowing the input vectors, we can rewrite the scalar matrix, by considering every scalar associated to the indeterminate z and rewriting them from left to right, starting from z's smallest power as:

$$G = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

This means that G has one input at the time, and two outputs.

Let us encode words of length 3.

Let us define the code's state Table 1.2, considering that the bits enter in the encoder from left to right:

This table shows what happens at a specific time t. The first cell of each line is the next bit going into the Linear Feedback Shift Register(LFSR), the first two cells show the sequence at time $t+1$ into both registers of the LFSR, right after the entry of the bit in the first cell; and the last two cells show the content of the LFSR at time t.

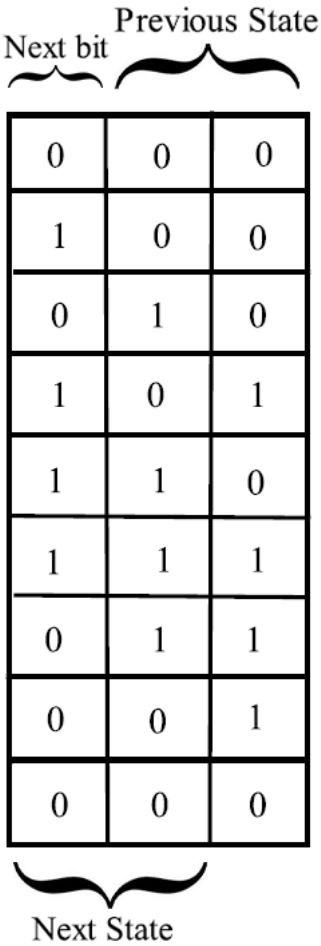

Figure 1.2: The State Table

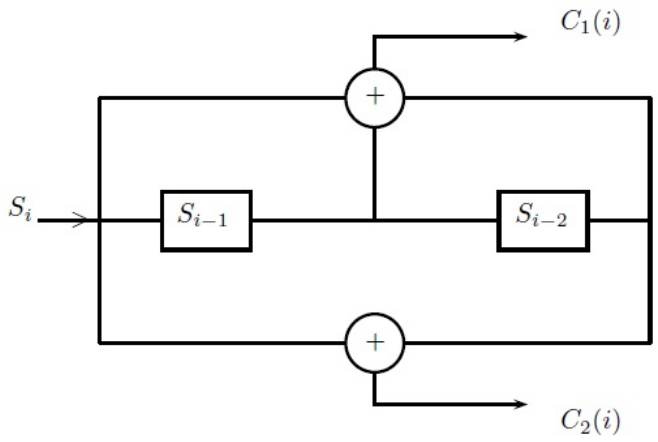

Figure 1.3: LFSR encoder

Now, let us have a look at both the LFSR encoder, and the state diagram. (A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state)

With the state diagram, we clearly see the possible states of the encoding machine, more precisely the various possible content of the registers. Moreover, it explains how to go from one state to another, and gives the output we get when moving from one state to a different one. Those outputs are the couples written on top of the long lines and underneath "Next bit". The state diagram is very helpful when encoding.

If we want to encode this set of words of length 3, considering that we always start with the state 00 this is how it goes.

In this example, we start from the state 00. As we move along, we send each and every bit of the sequence we wish to encode, from left to right. To compute the output sequence we get after every move, we take a look at the state diagram, and it gives us exactly the output value from one state to another, bit after bit.

From the generator matrix, we have both outputs:

$$\begin{cases} C_1(i) = & S_i + S_{i-1} + S_{i-2} \\ C_2(i) = & S_i + S_{i-2} \end{cases} \tag{1.6}$$

Then, we get the following codeword family:

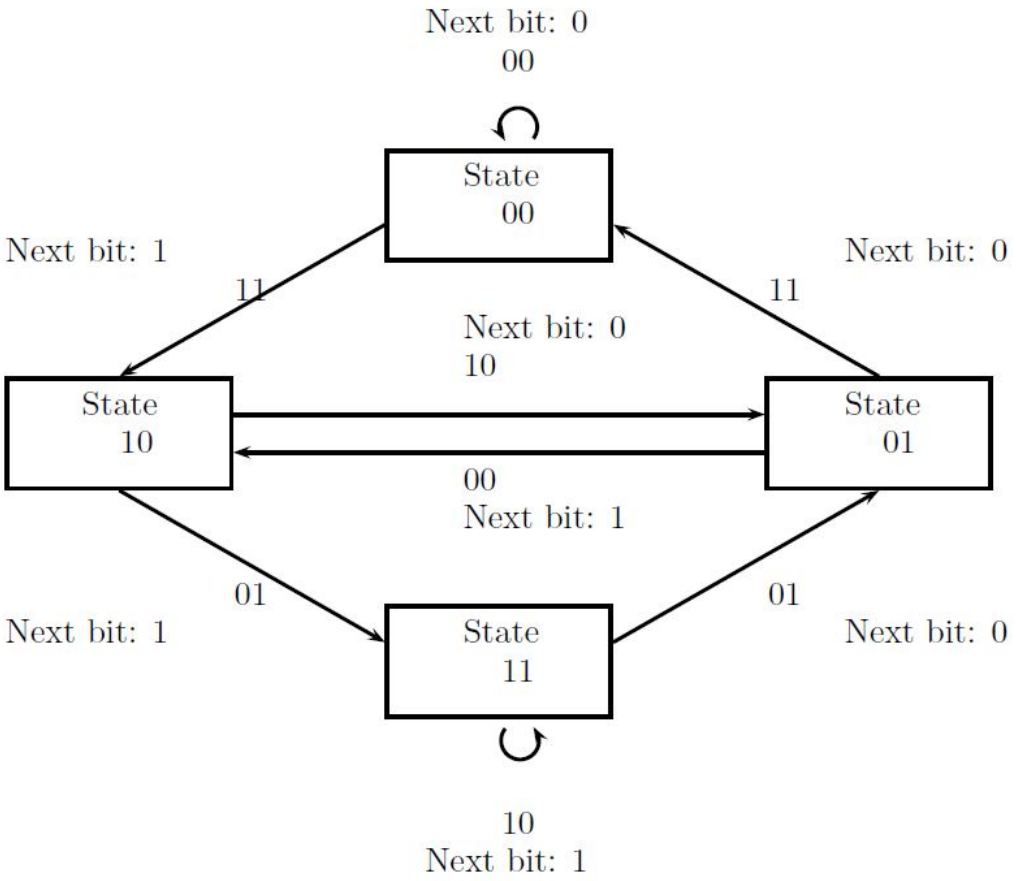

Figure 1.4: Diagram State

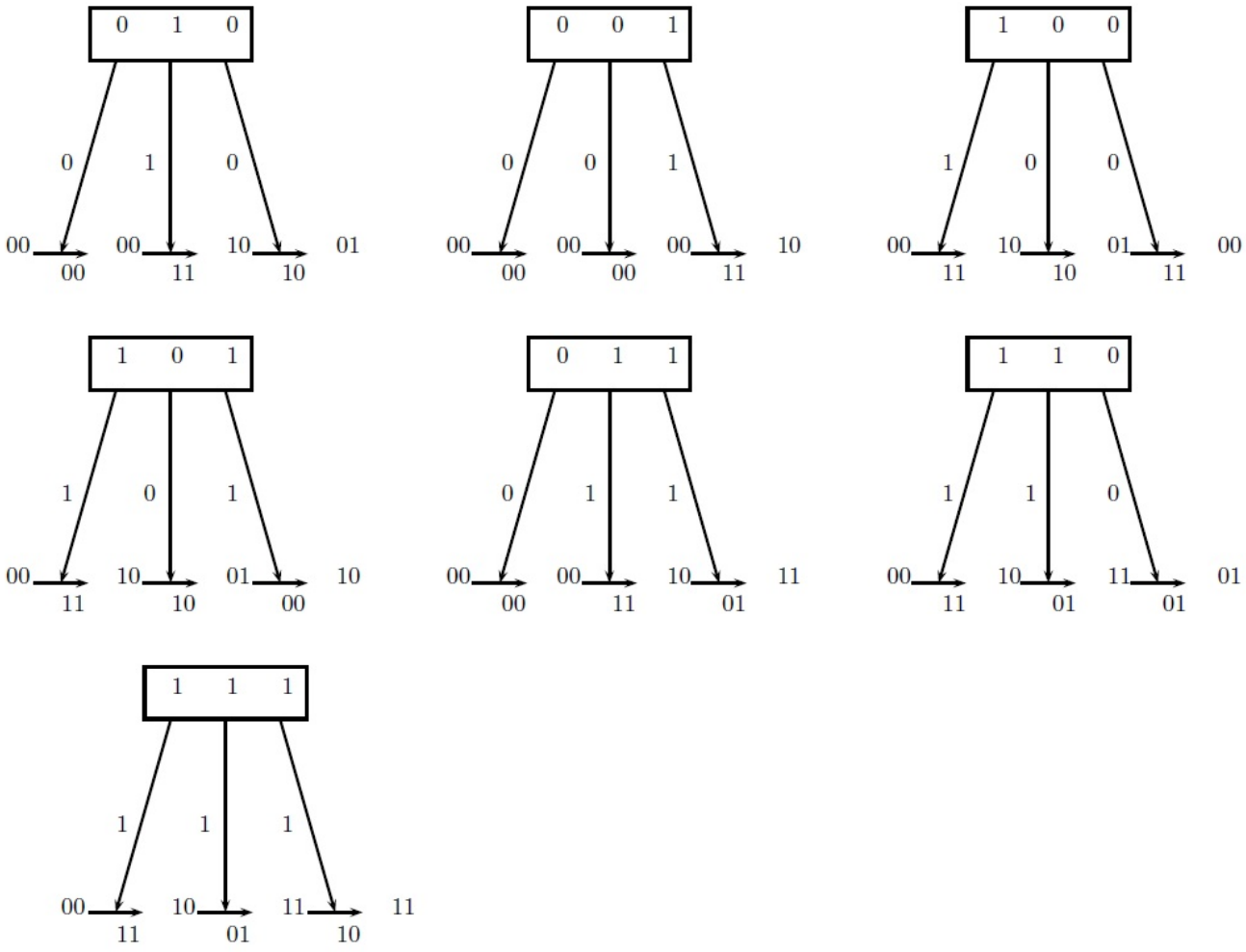

Figure 1.5: Encoding process

$$
\begin{array}{llllllll}
G(000, & 010, & 001, & 100, & 101, & 011, & 110, & 111) = \\
(000000, & 001110, & 000011, & 111011, & 111000, & 001101, & 110101, & 110110)
\end{array}
$$

This process is the convolutional way for encoding; which means that it is how semi-infinite sequences can be encoded bit after bit.
The free distance is given by the smallest distance to 000000 of every codeword; in our case it is $d(000011, 000000)$.
The free distance in this case is: 2.
Then, we have: $2 \leq 6$. $(k = 1, \delta = 2)$

Chapter 2

Systems theory

In this chapter, we recall the systems theory tools by introducing the input-state-output representation; then, we will talk about convolutional codes using the linear systems theory; and also introduce the realization for the transition between codes and linear systems. Then, we look at concatenated systems as linear systems, in order to later introduce the control properties of those systems focusing on their specific construction structure.

A discrete linear time-invariant system is described by the equations

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{2.1}$$

where $A \in M_\delta(\mathbb{F})$, $B \in M_{\delta\times k}(\mathbb{F})$, $C \in M_{p\times\delta}(\mathbb{F})$, $D \in M_{p\times k}(\mathbb{F})$ (with $p = n-k$) are constant matrices over the field $\mathbb{F}$, and $u(t) \in \mathbb{F}^k$, $x(t) \in \mathbb{F}^\delta$, $y(t) \in \mathbb{F}^p$ are the input, state and output vectors, respectively.

We will denote a system simply as the quadruple of matrices (A, B, C, D).

With initial condition $x(0) = 0$, a solution of the Equation (2.1) can be obtained by making use of the Z-transform. Let $u(z)$, $x(z)$, $y(z)$ be the Z-transforms of the variables u, x, y of a time- invariant linear system. Then by applying the Z-transform to the equations of the system we have

$$\begin{cases} zx(z) &= Ax(z) + Bu(z) \\ y(z) &= Cx(z) + Du(z) \end{cases} \tag{2.2}$$

and as a result we have

$$y(z) = (C(zI_\delta - A)^{-1}B + D)u(z), \tag{2.3}$$

called the transfer function of the system, and $C(zI_\delta - A)^{-1}B + D$ is the transfer matrix.

Remark 1. For any initial condition $x(0)$, the Z-transform takes the form

$$\begin{cases} zx(z) - x(0) &= Ax(z) + Bu(z) \\ y(z) &= Cx(z) + Du(z) \end{cases}$$

Remark 2. Observe that

$$C(zI_\delta - A)^{-1}B + D = \frac{1}{\det(zI_\delta - A)} C(Adj(zI_\delta - A))^t B + D.$$

Each entry of the adjoint matrix $Adj(zI_\delta - A)$ is a polynomial of degree strictly less than the degree of the determinant of $zI_\delta - A$. Consequently, $C(zI_\delta - A)^{-1}B$ is a strictly proper rational matrix, and if $D \neq 0$, $C(zI_\delta - A)^{-1}B + D$ is a proper rational matrix.

The values $z_0 \in \overline{\mathbb{F}}$ (where $\overline{\mathbb{F}}$ denotes the algebraic closure of the field $\mathbb{F}$) such that $\det(z_0 I_\delta - A) = 0$ are called eigenvalues of A and the set of all eigenvalues is called spectrum of A and is denoted by $Spec(A)$.

2.1 Convolutional codes as linear systems

Given a convolutional code, with a specific encoding matrix $G(z)$, we can always give a first-order representation of this matrix. By a series of transformations, we can find four matrices (A, B, C, D) of adequate sizes, corresponding to this matrix. The linear system (A, B, C, D) associated to the encoder $G(z)$ is called a realization of $G(z)$.

In general, such a representation is intended to have a different approach and understanding of codes, and be able to impact either the input, output, or the generator matrix, with the algebra material.

Linear systems for convolutional codes represent a mechanism to work on every little sub-piece of the encoding process. If we try to understand the physical control process, that goes along with the coding, the state of our encoding machine is modified by both the dynamics matrix and the input matrix. The final result, the output corresponds to the combination action of the sensor and the feedthrough matrices.

2.1.1 Realization algorithm

Let us consider some linear systems, that we would like to study as convolutional codes.

Given a linear system (A, B, C, D) it is easy to obtain a convolutional code such that the given system is a realization, as we can see in the following example.

Example 2.1.1. In $\mathbb{F}_7$, let us consider the realization (A, B, C, D) of a convolutional code with

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 1 & -3 \\ 1 & -2 \\ 0 & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 4 & 1 & 0 \end{pmatrix}, \; D = \begin{pmatrix} 1 & -2 \end{pmatrix}.$$

Indeed, we have the transfer matrix, which is given by the formula: $G(z) = C(zI_\delta - A)^{-1}B + D$.

We have: $(zI - A) = \begin{pmatrix} z & 0 & 0 \\ -4 & z & 1 \\ 0 & -1 & z \end{pmatrix}$. So,

$$(zI_\delta - A)^{-1} = \begin{pmatrix} \frac{1}{z} & 0 & 0 \\ \frac{4}{z^2+1} & \frac{z}{z^2+1} & \frac{-1}{z^2+1} \\ \frac{4}{z^3+z} & \frac{1}{z^2+1} & \frac{z}{z^2+1} \end{pmatrix}$$

and

$$\begin{aligned} & C(zI_\delta - A)^{-1}B + D \\ &= \begin{pmatrix} \frac{z^3+5z^2+5z+4}{z^3+z} & \frac{-2z^3-5}{z^3+z} \end{pmatrix}. \\ &= P(z)Q(z)^{-1} \end{aligned}$$

Taking $Q(z) = \begin{pmatrix} z^3+z & \\ & z^3+z \end{pmatrix}$, we consider the matrix

$$G(z)Q(z)^{-1} = \begin{pmatrix} P(z)Q(z)^{-1} \\ I \end{pmatrix}.$$

Then, the encoder $G(z)$ is given by the following matrix

$$G(z) = \begin{pmatrix} z^3 + 5z^2 + 5z + 4 & -2z^3 - 5 \\ z^3 + z & 0 \\ 0 & z^3 + z \end{pmatrix}$$

Equivalently, we can write as $\begin{pmatrix} z^3 + 5z^2 + 5z + 4 & 5z^3 + 2 \\ z^3 + z & 0 \\ 0 & z^3 + z \end{pmatrix}$

The convolutional code obtained out of this transformation can be simply designed by $\mathcal{C}(A, B, C, D)$.

We are interested in the inverse problem, that is to say, given a convolutional code we want to obtain a realization of this code.

From Theorem 1.2.1 and taking into account the following proposition

Proposition 2.1.1. *Let (K_1, L_1, M_1) be another representation of the convolutional code $\mathcal{C}$. Then there exist invertible matrices T and S of adequate sizes, such that*

$$(K_1, L_1, M_1) = (TKS^{-1}, TLS^{-1}, TM).$$

We have the following corollary

Corollary 2.1.1. *The triple (K, L, M) can be written as*

$$K = \begin{pmatrix} -I_\delta \\ 0 \end{pmatrix}, \; L = \begin{pmatrix} A \\ C \end{pmatrix}, \; M = \begin{pmatrix} 0 & B \\ -I_{n-k} & D \end{pmatrix}.$$

And we deduce the following corollary

Corollary 2.1.2.

$$\mathcal{C} = \{v(z) \in \mathbb{F}^n[z] \mid \exists x(z) \in \mathbb{F}^\delta[z] : \begin{pmatrix} zI - A & 0 & -B \\ -C & I & -D \end{pmatrix} \begin{pmatrix} x(z) \\ v(z) \end{pmatrix} = 0\}$$

Now, if we divide the matrix $v(z)$ in two parties $v(z) = \begin{pmatrix} y(z) \\ u(z) \end{pmatrix}$, the equality $\begin{pmatrix} zI - A & 0 & -B \\ -C & I & -D \end{pmatrix} \begin{pmatrix} x(z) \\ v(z) \end{pmatrix} = 0$ can be expressed as

$$\begin{cases} zx(z) &= Ax(z) + Bu(z) \\ y(z) &= Cx(z) + Du(z) \end{cases}$$

Finally, applying the antitransform Z, we obtain the system

$$\begin{cases} x(t+1) &= Ax(t)+Bu(t) \\ y(t) &= Cx(t)+Du(t) \end{cases}, \qquad \begin{aligned} v(t) &= \begin{pmatrix} u(t) \\ y(t) \end{pmatrix}, \\ x(0) &= 0. \end{aligned}$$

Remark 3. The vectors $u(t)$, $x(t)$, $y(t)$ and $v(t) = \begin{pmatrix} u(t) \\ y(t) \end{pmatrix}$ are known as information vector, state vector, parity vector and the code vector transmitted via the communication channel respectively.

Following the example 1.2.2, we have,

Example 2.1.2.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \sim$$

$$\begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 & 0 & -1 & 1 & 1 \end{pmatrix} \sim$$

$$\left(\begin{array}{ccc|ccc|c|cc} -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 2 & 1 & 0 & -1 & 1 & 1 \end{array}\right).$$

Then,

$$A = \begin{pmatrix} 0 & -1 & 0 \\ -1 & -1 & -1 \\ -1 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad .$$
$$C = \begin{pmatrix} 2 & 1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

Now, we present another method to obtain a realization.

Let $G(z)$ be a matrix generator of (n, k, δ) convolutional code, in which we consider that is in the form $\begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}$ with $Q(z)$ invertible and the degree δ of the polynomial $\det Q(z)$ being maximal among all minors of order k.

We decompose $P(z)Q(z)^{-1}$ into a polynomial matrix and a strictly proper matrix.

Let $p(z) = z^\delta + a_{\delta-1}z^{\delta-1} + \ldots + a_1 z + a_0$ the monic polynomial deduced from $\det Q(z)$. So, the matrix $P(z)Q(z)^{-1}$ is written in the form

$$\begin{pmatrix} d_{11} + \dfrac{q_{11}(z)}{p(z)} & \ldots & d_{1k} + \dfrac{q_{1k}(z)}{p(z)} \\ \vdots & & \vdots \\ d_{n-k1} + \dfrac{q_{n-k1}(z)}{p(z)} & \ldots & d_{n-kk} + \dfrac{q_{n-kk}(z)}{p(z)} \end{pmatrix}$$

$q_{ij} = c_0^{ij} + c_1^{ij} z + \ldots + c_{\delta-1}^{ij} z^{\delta-1}$

(by construction $d_{ij} \in \mathbb{F}$ and $\operatorname{degree} q_{ij} < \delta$).

First of all and for simplicity, we analyze the case where $k = 1$.

We consider the following matrices

$$D = \begin{pmatrix} d_{11} \\ \vdots \\ d_{n-k1} \end{pmatrix} \in M_{(n-k)\times 1}(\mathbb{F}).$$

$$A = \begin{pmatrix} -a_{\delta-1} & -a_{\delta-2} & \ldots & -a_1 & -a_0 \\ 1 & 0 & \ldots & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \in M_\delta(\mathbb{F})$$

$$B = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in M_{\delta\times 1}(\mathbb{F})$$

$$C = \begin{pmatrix} c_{\delta-1}^{11} & \cdots & c_0^{11} \\ \vdots & & \vdots \\ c_{\delta-1}^{n-k1} & \cdots & c_0^{n-k1} \end{pmatrix} \in M_{(n-k)\times\delta}.$$

A simple calculation shows that $C(zI_\delta - A)^{-1}B + D = P(z)Q(z)^{-1}$.

Example 2.1.3. Let $G(z)$ be the following encoder matrix

$$G(z) = \begin{pmatrix} 1+z+z^2 \\ 1+z^2 \end{pmatrix} = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}. \tag{2.4}$$

So,

$$C(zI - A)^{-1}B + D = P(z)Q(z)^{-1} = \frac{1+z+z^2}{1+z^2}$$

and we can decompose $P(z)Q(z)^{-1}$ into a polynomial matrix and a strictly proper matrix: $P(z)Q(z)^{-1} = 1 + \dfrac{z}{1+z^2}$. Then, we take the matrix D as the polynomial, and $C(zI - A)^{-1}B$ the strictly rational part.

So, $D = 1$ and $C(zI - A)^{-1}B = \dfrac{c_0 + c_1 z}{a_0 + a_1 z + z^2}$. Then $A = \begin{pmatrix} -a_1 & -a_0 \\ 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $C = \begin{pmatrix} c_1 & c_0 \end{pmatrix}$. So, $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, and $C = \begin{pmatrix} 1 & 0 \end{pmatrix}$.

Example 2.1.4. We consider the following code

$$G(z) = \begin{pmatrix} 1+z+z^2 \\ \alpha+z+\alpha^2 z^2 \\ \alpha^2+z+\alpha z^2 \end{pmatrix}$$

over the field $\mathbb{F}_4$,

First of all we make the addition and multiplication table of the field $\mathbb{F}_4$ considered.

$+$	0	1	α	$\alpha+1$
0	0	1	α	$\alpha+1$
1	1	0	$\alpha+1$	α
α	α	$\alpha+1$	0	1
$\alpha+1$	$\alpha+1$	α	1	0

$\cdot$	0	1	α	$\alpha+1$
0	0	0	0	0
1	0	1	α	$\alpha+1$
α	0	α	$\alpha+1$	1
$\alpha+1$	0	$\alpha+1$	1	α

(2.5)

$$G(z) = \begin{pmatrix} 1+z+z^2 \\ \alpha+z+\alpha^2 z^2 \\ \alpha^2+z+\alpha z^2 \end{pmatrix} = \begin{pmatrix} \dfrac{1+z+z^2}{\alpha^2+z+\alpha z^2} \\ \\ \dfrac{\alpha+z+\alpha^2 z^2}{\alpha^2+z+\alpha z^2} \\ \\ 1 \end{pmatrix} \alpha^2+z+\alpha z^2$$

$$\begin{pmatrix} \dfrac{1+z+z^2}{\alpha^2+z+\alpha z^2} \\ \\ \dfrac{\alpha+z+\alpha^2 z^2}{\alpha^2+z+\alpha z^2} \\ \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha+1+\dfrac{1+\alpha+\alpha z}{\alpha^2+z+\alpha z^2} \\ \\ \alpha+\dfrac{(1+\alpha)+(1+\alpha)z}{\alpha^2+z+\alpha z^2} \\ \\ 1 \end{pmatrix};$$

$$P(z)Q(z)^{-1} = \begin{pmatrix} 1+\alpha+\dfrac{\alpha+z}{\alpha+(\alpha+1)z+z^2} \\ \\ \alpha+\dfrac{\alpha+\alpha z}{\alpha+(\alpha+1)z+z^2} \end{pmatrix}.$$

Following as before we obtain the following realization (A, B, C, D) of the convolutional code where

$$D = \begin{pmatrix} \alpha+1 \\ \alpha \end{pmatrix},\ B = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$\begin{aligned} q_{11} &= \alpha+z = c_0^{11} + c_1^{11} z \\ q_{21} &= \alpha+\alpha z = c_0^{21} + c_1^{21} z \end{aligned}$$

$$C = \begin{pmatrix} c_1^{11} & c_0^{11} \\ c_1^{21} & c_0^{21} \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ \alpha & \alpha \end{pmatrix}$$

$$p(z) = a_0 + a_1 z + z^2 = \alpha + (1+\alpha)z + z^2$$

$$A = \begin{pmatrix} -a_1 & -a_0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1+\alpha & \alpha \\ 1 & 0 \end{pmatrix}$$

Now, we analyze the case $k > 1$

In this case, we construct the matrix A in the following manner

$$A = \begin{pmatrix} -a_{\delta-1}I_k & -a_{\delta-2}I_k & \dots & -a_1 I_k & -a_0 I_k \\ I_k & 0 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & I_k & 0 \end{pmatrix} \in M_{k\times\delta}(\mathbb{F})$$

now taking $D = \begin{pmatrix} d_{11} & \dots & d_{1k} \\ \vdots & & \vdots \\ d_{n-\delta 1} & \dots & d_{n-kk} \end{pmatrix}$, $B = \begin{pmatrix} I_k \\ 0_{k(\delta-1),k} \end{pmatrix}$ and $C = \begin{pmatrix} C_1 & \dots & C_\delta \end{pmatrix}$

where $C_1 = \begin{pmatrix} c_{\delta-1}^{11} & \dots & c_{\delta-1}^{1k} \\ \vdots & & \vdots \\ c_{\delta-1}^{n-k1} & \dots & c_{\delta-1}^{n-kk} \end{pmatrix}$, $C_2 = \begin{pmatrix} c_{\delta-2}^{11} & \dots & c_{\delta-2}^{1k} \\ \vdots & & \vdots \\ c_{\delta-2}^{n-k1} & \dots & c_{\delta-2}^{n-kk} \end{pmatrix}$, …,

$C_\delta = \begin{pmatrix} c_0^{11} & \dots & c_0^{1k} \\ \vdots & & \vdots \\ c_0^{n-k1} & \dots & c_0^{n-kk} \end{pmatrix}$.

Example 2.1.5. Let $G(z)$ be the following encoder matrix

$$G(z) = \begin{pmatrix} 1+z & 1 \\ z & 1+z \\ 1+z+z^2 & 0 \\ 0 & 1+z+z^2 \end{pmatrix} = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}$$

So,

$$C(zI-A)^{-1}B + D = P(z)Q(z)^{-1} = \begin{pmatrix} \dfrac{1+z}{1+z+z^2} & \dfrac{1}{1+z+z^2} \\ \dfrac{z}{1+z+z^2} & \dfrac{1+z}{1+z+z^2} \end{pmatrix}$$

In this case $D = 0$ and $A = \begin{pmatrix} -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$,

$C = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$.

2.2 Concatenated systems

For convolutional codes, concatenation is proved to improve the transmission. As an example, turbo codes, invented during the 90s, by an ENST-Bretagne group of researchers directed by Claude Berrou and Alain Glavieux, have been adopted by most of the telecommunication systems [5]. The turbo codes technique is built on the interlacing of recursive convolutional codes, concatenated in parallel. Systems concatenation are implemented in the same spirit.

In our work, we are going to consider two big families of concatenation: serial concatenation, and parallel concatenation and two other models of concatenation called systematic serial concatenation and parallel interleaver concatenation.

2.2.1 Serial concatenation

In the serial concatenation process, both codes are serialized one after another. Precisely, a first code is used to encode the information. This first codeword represents the second encoder's input, since it is then sent to the second box. The obtained word represents our final codeword.

Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be convolutional codes, called outer code, and inner code respectively. Let $x_1(t)$, $u_1(t)$, and $y_1(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and let $x_2(t)$, $u_2(t)$, and $y_2(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

The outer code $\mathcal{C}_1$ and the inner code $\mathcal{C}_2$ are serialized, one after the other, so that the input information $u_2(t) = y_1(t)$. Consequently

$$\begin{aligned} x_1(t+1) &= A_1 x_1(t) + B_1 u_1(t) \\ x_2(t+1) &= A_2 x_2(t) + B_2 C_1 x_1(t) + B_2 D_1 u_1(t) \\ y_2(t) &= C_2 x_2(t) + D_2 C_1 x_1(t) + D_2 D_1 u_1(t) \end{aligned}$$

That is to say the concatenated code is $\mathcal{C}(A, B, C, D)$ with

$$A = \begin{pmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix},$$
$$C = \begin{pmatrix} D_2C_1 & C_2 \end{pmatrix},\ D = D_2D_1. \tag{2.6}$$

If $\mathcal{C}_0(A_1, B_1, C_1, D_1)$ is a (m, k, δ_1)-code and $\mathcal{C}_i(A_2, B_2, C_2, D_2)$ is a $(n, m - k, \delta_2)$-code, then $\mathcal{C}(A, B, C, D)$ is a $(n - m + 2k, k, \delta_1 + \delta_2)$-code.

Example 2.2.1. In the field $\mathbb{F}_7$, let (A_1, B_1, C_1, D_1) with

$$A_1 = \begin{pmatrix} 1 & 0 & 2 \\ 5 & 1 & 5 \\ 4 & 3 & 0 \end{pmatrix},\ B_1 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 2 & 4 \end{pmatrix},$$
$$C_1 = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 1 & 3 \end{pmatrix},\ D_1 = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix} \tag{2.7}$$

and (A_2, B_2, C_2, D_2) with

$$A_2 = \begin{pmatrix} 2 & 4 \\ 5 & 1 \end{pmatrix},\ B_2 = \begin{pmatrix} 4 & 1 \\ 3 & 0 \end{pmatrix},$$
$$C_2 = \begin{pmatrix} 5 & 1 \\ 4 & 0 \\ 1 & 2 \end{pmatrix},\ D_2 = \begin{pmatrix} 5 & 1 \\ 2 & 6 \\ 0 & 4 \end{pmatrix} \tag{2.8}$$

be two realizations of two encoders $G_1(z)$ and $G_2(z)$. Then the realization of the serial concatenated is:

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 5 & 1 & 5 & 0 & 0 \\ 4 & 3 & 0 & 0 & 0 \\ 2 & 1 & 4 & 2 & 4 \\ 6 & 0 & 6 & 5 & 1 \end{pmatrix},\ B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 2 & 4 \\ 2 & 1 \\ 5 & 3 \end{pmatrix},$$
$$C = \begin{pmatrix} 4 & 1 & 6 & 5 & 1 \\ 3 & 6 & 1 & 4 & 0 \\ 4 & 4 & 5 & 1 & 2 \end{pmatrix},\ D = \begin{pmatrix} 6 & 2 \\ 1 & 5 \\ 0 & 2 \end{pmatrix}.$$

Proposition 2.2.1. *The transfer matrix defining the matrix encoder of the serial concatenated code is:*

$$G(z) = G_2(z)G_1(z) \tag{2.9}$$

where $G_1(z)$ and $G_2(z)$ are the transfer matrices corresponding to the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

Proof.

$$\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} = \begin{pmatrix} (zI_{\delta_1} - A_1)^{-1} & 0 \\ (zI_{\delta_2} - A_2)^{-1}B_2C_1(zI_{\delta_1} - A_1)^{-1} & (zI_{\delta_2} - A_2)^{-1} \end{pmatrix}.$$

So,

$$\begin{pmatrix} D_2C_1 & C_2 \end{pmatrix} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & sI_{\delta_2} - A_2 \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix} + D_2D_1 =$$

$D_2C_1(zI_{\delta_1} - A_1)^{-1}B_2C_1(zI_{\delta_1} - A_1)^{-1}B_1 + C_2(zI_{\delta_2} - A_2)^{-1}B_2D_1 + D_2D_1 = G_2(z)G_1(z)$.

□

2.2.2 Parallel concatenation

The second concatenated model that we will study is the parallel concatenation. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be convolutional codes. Let $x_1(t)$, $u_1(t)$, and $y_1(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and let $x_2(t)$, $u_2(t)$, and $y_2(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

Both codes are concatenated in a parallel form, so that the input information is $u_2(t) = u_1(t) = u(t)$ and the final parity vector is $y(t) = y_1(t) + y_2(t)$. Consequently

$$\begin{aligned} x_1 &= A_1x_1(t) + B_1u(t) \\ x_2 &= A_2x_2(t) + B_2u(t) \\ y(t) &= C_1x_1(t) + C_2x_2(t) + (D_1 + D_2)u(t) \end{aligned}$$

$$\begin{aligned} A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \\ C = \begin{pmatrix} C_1 & C_2 \end{pmatrix},\ D = D_1 + D_2. \end{aligned} \tag{2.10}$$

If $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ is a (n, k, δ_1)-code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ is a (n, k, δ_2)-code, then $\mathcal{C}(A, B, C, D)$ is a $(n, k, \delta_1 + \delta_2)$-code.

Example 2.2.2. In the field $\mathbb{F}_7$, let (A_1, B_1, C_1, D_1) with

$$\begin{aligned} A_1 &= \begin{pmatrix} 1 & 3 \\ 4 & 1 \end{pmatrix}, \ B_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \\ C_1 &= \begin{pmatrix} 5 & 2 \\ 0 & 6 \\ 3 & 0 \end{pmatrix}, \ D_1 = \begin{pmatrix} 2 \\ 5 \\ 6 \end{pmatrix}. \end{aligned} \tag{2.11}$$

and (A_2, B_2, C_2, D_2) with

$$\begin{aligned} A_2 &= \begin{pmatrix} 6 & 3 & 0 \\ 0 & 6 & 2 \\ 1 & 0 & 1 \end{pmatrix}, \ B_2 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \\ C_2 &= \begin{pmatrix} 4 & 2 & 6 \\ 6 & 5 & 5 \\ 3 & 3 & 2 \end{pmatrix}, \ D_2 = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix}. \end{aligned} \tag{2.12}$$

be two realizations of two encoders $G_1(z)$ and $G_2(z)$. Then the realization of the parallel concatenated is:

$$\begin{aligned} A &= \begin{pmatrix} 1 & 3 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 6 & 3 & 0 \\ 0 & 0 & 0 & 6 & 2 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \ B = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 2 \\ 3 \end{pmatrix}, \\ C &= \begin{pmatrix} 5 & 2 & 4 & 2 & 6 \\ 0 & 6 & 6 & 5 & 5 \\ 3 & 0 & 3 & 3 & 2 \end{pmatrix}, \ D = \begin{pmatrix} 3 \\ 5 \\ 3 \end{pmatrix}. \end{aligned}$$

Proposition 2.2.2. *The transfer matrix defining the matrix encoder of the parallel concatenated code is:*

$$G(z) = G_1(z) + G_2(z) \tag{2.13}$$

where $G_1(z)$ and $G_2(z)$ are the transfer matrices corresponding to the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

Proof.

$$\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ 0 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} = \begin{pmatrix} (zI_{\delta_1} - A_1)^{-1} & 0 \\ 0 & (zI_{\delta_2} - A_2)^{-1} \end{pmatrix}.$$

So,

$$\begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ 0 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} + D_1 + D_2 =$$
$$C_1(zI_{\delta_1} - A_1)^{-1}B_1 + D_1 + C_2(zI_{\delta_2} - A_2)^{-1}B_2 + D_2 =$$
$$G_1(z) + G_2(z).$$

□

2.2.3 Systematic serial concatenation

In this case we consider the same kind of codes than the case of serial concatenation. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be convolutional codes, called outer code, and inner code respectively. Let $x_1(t)$, $u_1(t)$, and $y_1(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and let $x_2(t)$, $u_2(t)$, and $y_2(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

The systematic serial concatenation of these codes is $\mathcal{C}(A, B, C, D)$ with

$$A = \begin{pmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix},\ C = \begin{pmatrix} C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix},\ D = \begin{pmatrix} D_1 \\ D_2D_1 \end{pmatrix}.$$

If $\mathcal{C}_0(A_1, B_1, C_1, D_1)$ is a (m, k, δ_1)-code and $\mathcal{C}_i(A_2, B_2, C_2, D_2)$ is a $(n, m-k, \delta_2)$-code, then $\mathcal{C}(A, B, C, D)$ is a $(n+k, k, \delta_1+\delta_2)$-code.

Example 2.2.3. In the field $\mathbb{F}_7$, let (A_1, B_1, C_1, D_1) with

$$\begin{aligned} A_1 = \begin{pmatrix} 1 & 0 & 2 \\ 5 & 1 & 5 \\ 4 & 3 & 0 \end{pmatrix},\ B_1 = \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 2 & 4 \end{pmatrix}, \\ C_1 = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 1 & 3 \end{pmatrix},\ D_1 = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix} \end{aligned} \tag{2.14}$$

and (A_2, B_2, C_2, D_2) with

$$\begin{aligned} A_2 = \begin{pmatrix} 2 & 4 \\ 5 & 1 \end{pmatrix},\ B_2 = \begin{pmatrix} 4 & 1 \\ 3 & 0 \end{pmatrix}, \\ C_2 = \begin{pmatrix} 5 & 1 \\ 4 & 0 \\ 1 & 2 \end{pmatrix},\ D_2 = \begin{pmatrix} 5 & 1 \\ 2 & 6 \\ 0 & 4 \end{pmatrix} \end{aligned} \tag{2.15}$$

be two realizations of two encoders $G_1(z)$ and $G_2(z)$. Then the realization of the systematic serial concatenated is:

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 5 & 1 & 5 & 0 & 0 \\ 4 & 3 & 0 & 0 & 0 \\ 2 & 1 & 4 & 2 & 4 \\ 6 & 0 & 6 & 5 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 2 & 4 \\ 2 & 1 \\ 5 & 3 \end{pmatrix},$$
$$C = \begin{pmatrix} 2 & 0 & 2 & 0 & 0 \\ 1 & 1 & 3 & 0 & 0 \\ 4 & 1 & 6 & 5 & 1 \\ 3 & 6 & 1 & 4 & 0 \\ 4 & 4 & 5 & 1 & 2 \end{pmatrix}, \; D = \begin{pmatrix} 4 & 1 \\ 0 & 4 \\ 6 & 2 \\ 1 & 5 \\ 0 & 2 \end{pmatrix}.$$

Proposition 2.2.3. *The transfer matrix defining the matrix encoder of the systematic serial concatenated code is*

$$G(z) = \begin{pmatrix} G_1(z) \\ G_2(z)G_1(z) \end{pmatrix} \tag{2.16}$$

where $G_1(z)$ and $G_2(z)$ are the transfer matrices corresponding to the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

Proof.

$$\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} = \begin{pmatrix} (zI_{\delta_1} - A_1)^{-1} & 0 \\ (zI_{\delta_2} - A_2)^{-1}B_2C_1(zI_{\delta_1} - A_1)^{-1} & (zI_{\delta_2} - A_2)^{-1} \end{pmatrix}.$$

So,

$$\begin{pmatrix} C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & sI_{\delta_2} - A_2 \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix} + \begin{pmatrix} D_1 \\ D_2D_1 \end{pmatrix}$$
$$= \begin{pmatrix} C_1(zI_{\delta_1} - A_1)^{-1}B_1 + D_1 \\ D_2C_1(zI_{\delta_1} - A_1)^{-1}B_2C_1(zI_{\delta_1} - A_1)^{-1}B_1 + C_2(zI_{\delta_2} - A_2)^{-1}B_2D_1 + D_2D_1 \end{pmatrix}$$
$$= \begin{pmatrix} G_1(z) \\ G_2(z)G_1(z) \end{pmatrix}.$$

□

2.2.4 Parallel Interleaver concatenation

The other concatenated model that we will study is the parallel concatenation with interleaver. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ be a (n, k, δ_1)-convolutional code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be an (n, k, δ_2)-convolutional code, and $\mathcal{P}$ a $(k \times k)$ permutation matrix that We call the interleaver matrix. Let $x_1(t)$, $u_1(t)$, and $y_1(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and let $x_2(t)$, $u_2(t)$, and $y_2(t)$ be the state vector, the information vector and the parity vector of $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively. We suppose that the input sequence goes through an interleaver before being encoded with the encoder $\mathcal{C}_2$. We are going to encode in parallel on one side the first code $\mathcal{C}_1$, and on the other side, the code $\mathcal{C}_2(A_2, \bar{B}_2, C_2, \bar{D}_2)$ with $\bar{B}_2 = B_2\mathcal{P}$ and $\bar{D}_2 = D_2\mathcal{P}$ obtained after to apply the interleaver to the second code $\mathcal{C}_2$.

Both sides are concatenated in a parallel form, so that the input information $u_1(t) = u(t)$, and $u_2(t) = \mathcal{P}u(t)$ and the final parity vector $y(t) = y_1(t) + \bar{y_2}(t)$. Consequently, we get the new realization:

$$\begin{cases} x_1(t) &= A_1x_1(t) + B_1u(t) \\ x_2(t) &= A_2x_2(t) + B_2\mathcal{P}u(t) \\ y(t) &= C_1x_1(t) + C_2x_2(t) + (D_1 + D_2\mathcal{P})u(t) \end{cases}$$

$$\begin{aligned} A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2\mathcal{P} \end{pmatrix}, \\ C = \begin{pmatrix} C_1 & C_2 \end{pmatrix},\ D = D_1 + D_2\mathcal{P}. \end{aligned} \tag{2.17}$$

If $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ is a (n, k, δ_1)-code and $\mathcal{C}_2(A_2, \bar{B}_2, C_2, \bar{D}_2)$ is a (n, k, δ_2)-code, then $\mathcal{C}(A, B, C, D)$ is a $(n, k, \delta_1 + \delta_2)$-code.

Example 2.2.4. In $\mathbb{F}_5$, let us consider (A_1, B_1, C_1, D_1) with

$$\begin{aligned} A_1 = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix},\ B_1 = \begin{pmatrix} 4 & 3 \\ 1 & 0 \\ 2 & 2 \end{pmatrix}, \\ C_1 = \begin{pmatrix} 2 & 1 & 0 \end{pmatrix},\ D_1 = \begin{pmatrix} 2 & 0 \end{pmatrix} \end{aligned} \tag{2.18}$$

and (A_2, B_2, C_2, D_2) with

$$A_2 = \begin{pmatrix} 4 & 2 & 1 \\ 3 & 0 & 3 \\ 1 & 2 & 1 \end{pmatrix}, \; B_2 = \begin{pmatrix} 1 & 3 \\ 0 & 2 \\ 2 & 1 \end{pmatrix}, \\ C_2 = \begin{pmatrix} 0 & 3 & 4 \end{pmatrix}, \; D_2 = \begin{pmatrix} 4 & 2 \end{pmatrix} \tag{2.19}$$

and the interleaver matrix $\mathcal{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

The concatenated system (A, B, C, D) is such that:

$$A = \begin{pmatrix} 0 & 1 & 2 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 1 \\ 0 & 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 1 & 2 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 4 & 3 \\ 1 & 0 \\ 2 & 2 \\ 3 & 1 \\ 2 & 0 \\ 1 & 2 \end{pmatrix}, \\ C = \begin{pmatrix} 2 & 1 & 0 & 0 & 3 & 4 \end{pmatrix}, \; D = \begin{pmatrix} 4 & 4 \end{pmatrix}. \tag{2.20}$$

Proposition 2.2.4. *The transfer matrix defining the matrix encoder of the concatenated code with interleaver is*

$$G(z) = G_1(z) + G_2(z)\mathcal{P}, \tag{2.21}$$

where $G_1(z)$ and $G_2(z)$ are the transfer matrices corresponding to the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, respectively.

Proof.

$$\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ 0 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} = \begin{pmatrix} (zI_{\delta_1} - A_1)^{-1} & 0 \\ 0 & (zI_{\delta_2} - A_2)^{-1} \end{pmatrix}.$$

So,

$$\begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ 0 & zI_{\delta_2} - A_2 \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2\mathcal{P} \end{pmatrix} + (D_1 + D_2\mathcal{P}) =$$

$$\begin{pmatrix} C_1(zI_{\delta_1} - A_1)^{-1}B_1 + D_1 \end{pmatrix} + \begin{pmatrix} C_2(zI_{\delta_2} - A_2)^{-1}B_2\mathcal{P} + D_2\mathcal{P} \end{pmatrix} = \\ G_1(z) + G_2(z)\mathcal{P}.$$

□

Chapter 3

Controllability and Observability

In control systems theory the major concepts are controllability and observability. These concepts were introduced by R. Kalman in 1960 ([46]). Roughly speaking observability means the possibility of identifying the internal state of a system from measurements of the outputs. Controllability means instead the possibility of steering the system from any initial state to any final one by means of a control signal in the input. In this chapter, we recall some properties and notions of the control theory, such as controllability and observability, and we will introduce output-observability as well. We want to remark that in Control Theory the output observability is known as functional output-controllability that generally means, that the system can steer output of dynamical system along the arbitrary given curve over any interval of time, independently of its state vector. Nevertheless, the name used in coding theory is to output-observability and for that we use this name in this thesis.

In this chapter, we write those properties for concatenated systems; we will study necessary or sufficient conditions in order to reach those properties for concatenated codes. We highlight that it contains in majority some of the articles we published.
Let us formalize these concepts.

3.1 Controllability, observability

In this section, we introduce the definitions, as we use those results and get inspired by the concepts, since they are very useful for the rest of the thesis.

Definition 3.1.1. A linear system (A, B, C, D) is a controllable system if the controllability matrix of the system

$$\mathcal{C} = \begin{pmatrix} B & AB & A^2B & \dots & A^{\delta-1}B \end{pmatrix} \tag{3.1}$$

has full rank δ, where δ is the complexity of the code.

Or equivalently (Hautus test [38]), a linear system (A, B, C, D) is controllable if and only if

$$\text{rank } \begin{pmatrix} zI - A & B \end{pmatrix} = \delta, \text{ for all } z \in \overline{\mathbb{F}}, \tag{3.2}$$

where $\overline{\mathbb{F}}$ denotes the algebraic closure of $\mathbb{F}$.

Definition 3.1.2. A linear system (A, B, C, D) is said to be observable if the observability matrix of the system

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\delta-1} \end{pmatrix} \tag{3.3}$$

has full rank δ.

or equivalently (Hautus test [38]), a linear system (A, B, C, D) is observable if and only if

$$\text{rank } \begin{pmatrix} zI - A \\ C \end{pmatrix} = \delta, \text{ for all } z \in \overline{\mathbb{F}}, \tag{3.4}$$

where $\overline{\mathbb{F}}$ denotes the algebraic closure of $\mathbb{F}$.

Example 3.1.1. Over $\mathbb{F} = \mathbb{Z}_2$, we consider the encoder

$$G(z) = \begin{pmatrix} 1 + z + z^2 \\ z + z^2 + z^3 \\ z^2 + z^3 + z^4 \\ z^3 + z^4 + z^5 \end{pmatrix} = \begin{pmatrix} \frac{1+z+z^2}{z^3+z^4+z^5} \\ \\ \frac{z+z^2+z^3}{z^3+z^4+z^5} \\ \\ \frac{z^2+z^3+z^4}{z^3+z^4+z^5} \\ \\ 1 \end{pmatrix} (z^3 + z^4 + z^5).$$

corresponding to the encoder of a convolutional code A realization of this code is (A, B, C, D) where $A = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $C = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$, and $D = 0_{3\times 1}$.

Taking into account that

$$\operatorname{rank}\begin{pmatrix} B & AB & A^2B & A^3B & A^4B \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & -1 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = 5$$

and

$$\operatorname{rank}\begin{pmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = 3 < 5,$$

this system is controllable but not observable.

For controllable systems, the following result is well known and that we reproduce by the interest for this work.

Proposition 3.1.1. *If a system is controllable, then from the initial state $x_0 = 0$ we can reach any state $x \in \mathbb{F}^\delta$.*

Proof. If $\operatorname{rank}\begin{pmatrix} B & AB & \dots & A^{\delta-1}B \end{pmatrix} = \delta$, for any vector x there exists $u(0)$,

$u(1)$, ..., $u(\delta-1)$ such that

$$x = Bu(0) + ABu(1) + \ldots + A^{\delta-1}Bu(\delta-1)$$

Then, from $x_0 = 0$ taking the inputs $u(0)$, $u(1)$, ..., $u(\delta-1)$ we have

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) = Bu(0) \\ x(2) &= Ax(1) + Bu(1) = ABu(0) + Bu(1) \\ x(3) &= A^2Bu(0) + ABu(1) + Bu(2) \\ &\vdots \\ x(\delta) &= A^{\delta-1}Bu(0) + \ldots + ABu(\delta-2) + Bu(\delta-1) = x \end{aligned}$$

□

The observability character of a code means that one can be sure that a message has been completed once a sufficiently long string of zeros has been received.

Proposition 3.1.2 ([71])**.** *A code C (on $\mathbb{Z}_+$) is observable if and only if it has a generator matrix $G(z)$ (and therefore any other generator matrix) that is right-prime when is considered as a matrix over $\mathbb{F}[z, z^{-1}]$. That is to say their minors of size $k \times k$ are non-zero and have no common trivial factors (considering factors z^n with $n \in \mathbb{N}$ as trivial). If $G(z)$ is a generator matrix of a convolutional code observable, then $G(z)$ is not catastrophic (More information can be found in [61])*

3.2 Output-observability

Related to the minimality realization of an encoder is the output-observability property.

Output-observability represents the possibility of an internal state, to be only defined by a finite set of outputs, for a finite number of steps. In their work [30], Mª I. García-Planas, and S. Tarragona presented this concept, from a general point of view. Indeed, the definition is given for singular linear systems, over $\mathbb{C}$, but we can apply it to our convolutional coding context, where the support spaces are finite fields. Our approach is very inspired by theirs.

Definition 3.2.1. A system (A, B, C, D) is said to be output observable if the state sequence $x(0), \ldots, x(\ell)$ is uniquely determined by the knowledge of the output sequence $y(0), \ldots, y(\ell)$ for a finite number of steps $\ell \in \mathbb{N}$.

Observe that $x(1), \ldots, x(\ell)$ are determined by the knowledge of $x(0)$ and $u(0), \ldots, u(\ell-1)$ because of

$$\begin{cases} x(1) &= Ax(0) + Bu(0) \\ x(2) &= Ax(1) + Bu(1) \\ &= A^2x(0) + ABu(0) + Bu(1) \\ &\vdots \\ x(\ell) &= Ax(\ell-1) + Bu(\ell-1) \\ &= A^\ell x(0) + A^{\ell-1}Bu(0) + \ldots + ABu(\ell-2) + Bu(\ell-1), \end{cases}$$

and the elements $x(0), u(0), \ldots,$ and $u(\ell)$ can be obtained solving the following system of matrix equations.

$$\begin{cases} y(0) &= Cx(0) + Du(0) \\ y(1) &= Cx(1) + Du(1) \\ &= CAx(0) + CBu(0) + Du(1) \\ &\vdots \\ y(\ell) &= Cx(\ell) + Du(\ell) \\ &= CA^\ell x(0) + CA^{\ell-1}Bu(0) + \ldots + CBu(\ell-1) + Du(\ell). \end{cases} \tag{3.5}$$

In a more general way we can define the output-observability character saying that the state sequence $x(s), \ldots, x(\ell)$ is uniquely determined by the knowledge of the output sequence $y(s), \ldots, y(s+\ell)$ for a finite number of steps $\ell \in \mathbb{N}$.

In an analogous way we have that $x(s+1), \ldots, x(s+\ell)$ are determined by the knowledge of $x(s)$ and $u(s), \ldots, u(s+\ell)$ because

$$\begin{cases} x(s+1) &= Ax(s) + Bu(s) \\ x(s+2) &= Ax(s+1) + Bu(s+1) \\ &= A^2x(s) + ABu(s) + Bu(s+1) \\ &\vdots \\ x(s+\ell) &= Ax(s+\ell-1) + Bu(s+\ell-1) \\ &= A^\ell x(s) + A^{\ell-1}Bu(s) + \ldots + ABu(s+\ell-2) + Bu(s+\ell-1), \end{cases}$$

and the elements $x(s)$, and $u(s), \ldots, u(s+\ell)$ can be obtained by solving the

following system of matrix equations.

$$\begin{cases} y(s) &= Cx(s) + Du(s) \\ y(s+1) &= Cx(s+1) + Du(s+1) \\ &= CAx(s) + CBu(s) + Du(s+1) \\ &\vdots \\ y(s+\ell) &= Cx(s+\ell) + Du(s+\ell) \\ &= CA^{\ell}x(s) + CA^{\ell-1}Bu(s) + \ldots + CBu(s+\ell-1) + Du(s+\ell). \end{cases} \tag{3.6}$$

Calling $T_\ell(A, B, C, D)$ (that we simply write T_ℓ if no confusion is possible) the matrix

$$T_\ell = \begin{pmatrix} C & D & & & & \\ CA & CB & D & & & \\ CA^2 & CAB & CB & D & & \\ \vdots & & & \ddots & \ddots & \\ CA^{\ell} & CA^{\ell-1}B & CA^{\ell-2}B & \ldots & CB & D \end{pmatrix}. \tag{3.7}$$

We have the following.

Proposition 3.2.1. *A system (A, B, C, D) is output observable if and only if the matrix T_ℓ has full row rank for all $\ell \in \mathbb{N}$.*

Proof. It suffices to observe that for each ℓ, the matrix T_ℓ is the corresponding matrix to the system (3.5).

□

Remark 4. If the number of rows is bigger than the number of columns, there are values of $y(0), \ldots, y(\ell)$, for which $(y(0), \ldots, y(\ell))$ is not a parity vector.

Corollary 3.2.1. *A necessary condition for output-observability of the system (A, B, C, D) is that the matrix $\begin{pmatrix} C & D \end{pmatrix}$ has full row rank.*

Example 3.2.1. Let α be a primitive element of $\mathbb{F} = GF(8) = \mathbb{Z}_2[\alpha]/\alpha^3+\alpha^2+1$.

Let (A, B, C, D) be a realization of a convolutional code with

$$A = \begin{pmatrix} \alpha^3 & \alpha^2 \\ \alpha & 1 \end{pmatrix},\ B = \begin{pmatrix} 1 & \alpha \\ 1 & \alpha^2 \end{pmatrix},\ C = \begin{pmatrix} 1 & \alpha^6 \\ \alpha^3 & \alpha^2 \end{pmatrix},\ D = \begin{pmatrix} \alpha^2 & \alpha^4 \\ \alpha^5 & 1 \end{pmatrix}.$$

The system is not output observable because

$$\operatorname{rank}\begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & \alpha^6 & \alpha^2 & \alpha^4 \\ \alpha^3 & \alpha^2 & \alpha^5 & 1 \end{pmatrix} = 1 < 2.$$

Therefore, we assume that the number of rows is less than or equal to the number of columns. It is well known that in this case and for each ℓ, the systems (3.5) have solution for all $y(0), \ldots, y(\ell)$, if and only if the systems have full rank.

The following corollary is obvious.

Corollary 3.2.2. *If the matrix D in the system (A, B, C, D) has full row rank, then the system is output-observable.*

Fixing the initial state $x(s) = 0$, the output-observability matrix allows us to describe a sequence of trajectories $\{v_s, \ldots, v_{s+\ell}\}$ in the following manner.

Theorem 3.2.1. *Let (A, B, C, D) be a representation of a convolutional code. Suppose that the initial state of the system is $x(s) = 0$, then*

$$\{v_s, \ldots, v_{s+\ell}\} = \operatorname{Ker} \mathcal{T}_\ell,$$

where

$$\mathcal{T}_\ell = \begin{pmatrix} D & -I & & & & & & & \\ CB & 0 & D & -I & & & & & \\ CAB & 0 & CB & 0 & D & -I & & & \\ \vdots & & \ddots & \ddots & & & & & \\ CA^{\ell-1}B & 0 & CA^{\ell-2}B & 0 & \ldots & CB & 0 & D & -I \end{pmatrix}$$

Proof. The system

$$T_\ell \begin{pmatrix} x(s) \\ u(s) \\ \vdots \\ u(s+\ell) \end{pmatrix} = \begin{pmatrix} y(s) \\ \vdots \\ y(s+\ell) \end{pmatrix}$$

is equivalent to

$$\begin{pmatrix} T_\ell & -I \end{pmatrix} \begin{pmatrix} x(s) \\ u(s) \\ \vdots \\ u(s+\ell) \\ y(s) \\ \vdots \\ y(s+\ell) \end{pmatrix} = 0.$$

And now, it suffices to make column block elementary transformations to the system matrix. □

3.2.1 Alternative method for output-observability

Now, we present a new and simple method to analyze output-observability character. This method is simple in the sense that we do not need to compute the products of the matrices CA^iB.

Let (A, B, C, D) be a system and we consider the matrices that we will write $M_\ell(A, B, C, D)$ (that we simply write M_ℓ if confusion is not possible) defined in the following manner:

$$M_\ell = \begin{pmatrix} A & B & -I & 0 & 0 & 0 & \dots & & 0 & 0 \\ C & D & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & A & B & -I & 0 & & & & \\ 0 & 0 & C & D & 0 & 0 & & & & \\ \vdots & & & & & \ddots & & & & \\ & & & \dots & & & A & B & -I & 0 \\ & & & \dots & & & C & D & 0 & 0 \\ & & & \dots & & & 0 & 0 & C & D \end{pmatrix} \tag{3.8}$$

where $M_\ell \in M_{(\ell(\delta+p)+p)\times(\ell+1)(\delta+k)}(\mathbb{F})$.

We have the following result.

Theorem 3.2.2. *Let* (A, B, C, D) *be a system. Then*

$$\operatorname{rank} M_\ell = \operatorname{rank} (T_\ell + \ell\delta).$$

Proof. Making block row and column elementary transformations, we have

$$\operatorname{rank}\begin{pmatrix} A & B & -I & 0 & 0 & 0 & \dots & & 0 & 0 \\ C & D & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & A & B & -I & 0 & & & & \\ 0 & 0 & C & D & 0 & 0 & & & & \\ \vdots & & & & & \ddots & & & & \\ & & & \dots & & & A & B & -I & 0 \\ & & & \dots & & & C & D & 0 & 0 \\ & & & \dots & & & 0 & 0 & C & D \end{pmatrix}$$

$$= \operatorname{rank}\begin{pmatrix} I & & & & & & & & \\ & \overset{(\ell)}{\ddots} & & & & & & & \\ & & I & & & & & & \\ & & & C & D & & & & \\ & & & CA & CB & D & & & \\ & & & CA^2 & CAB & CB & D & & \\ & & & \vdots & & \ddots & & & \\ & & & CA^\ell & CA^{\ell-1}B & & CB & D \end{pmatrix}.$$

□

In order to obtain properties, we define the following equivalence relation preserving the required properties.

Definition 3.2.2. The systems (A, B, C, D) and (A_1, B_1, C_1, D_1) are feedback equivalent, that we write

$$(A, B, C, D) \sim (A_1, B_1, C_1, D_1),$$

if and only if

$$\begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} = \begin{pmatrix} P^{-1} & W \\ 0 & S \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} P & 0 \\ V & R \end{pmatrix}, \tag{3.9}$$

for some matrices $P \in M_\delta(\mathbb{F})$, $R \in M_k(\mathbb{F})$, $S \in M_p(\mathbb{F})$, $V \in M_{k\times\delta}(\mathbb{F})$ and $W \in M_{\delta\times p}(\mathbb{F})$.

Remark 5. Note that this equivalence generalizes the similarity equivalence

$$(A, B, C, D) \simeq (A_1, B_1, C_1, D_1)$$

if and only if

$$(A_1, B_1, C_1, D_1) = (P^{-1}AP, P^{-1}B, CP, D)$$

It suffices to take $V = 0$, $W = 0$, $R = I_m$, and $S = I_p$.

Proposition 3.2.2. *Let (A, B, C, D) and (A_1, B_1, C_1, D_1) be equivalent systems under equivalence relation considered. Then*

$$\operatorname{rank} M_\ell(A, B, C, D) = \operatorname{rank} M_\ell(A_1, B_1, C_1, D_1), \tag{3.10}$$

for all $\ell \in \mathbb{N}$.

Proof. Calling

$$\mathbf{P} = \begin{pmatrix} P^{-1} & W & & & & & & \\ 0 & S & & & & & & \\ 0 & 0 & P^{-1} & W & & & & \\ 0 & 0 & 0 & S & & & & \\ & & & & \ddots & & & \\ & & & & & P^{-1} & W & \\ & & & & & 0 & S & \\ & & & & & & & S \end{pmatrix},$$

and

$$\mathbf{Q} = \begin{pmatrix} P & 0 & & & & & & \\ V & R & & & & & & \\ & & P & 0 & & & & \\ & & V & R & & & & \\ & & & & \ddots & & & \\ & & & & & P & 0 & \\ & & & & & V & R & \\ & & & & & & & R \\ & & & & & & & & R \end{pmatrix},$$

We have

$$\mathbf{P}\begin{pmatrix} A & B & -I & 0 & 0 & 0 & \dots & & 0 & 0 \\ C & D & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & A & B & -I & 0 & & & & \\ 0 & 0 & C & D & 0 & 0 & & & & \\ \vdots & & & & & \ddots & & & & \\ & & & \dots & & & A & B & -I & 0 \\ & & & \dots & & & C & D & 0 & 0 \\ & & & \dots & & & 0 & 0 & C & D \end{pmatrix}\mathbf{Q} =$$

$$\begin{pmatrix} A_1 & B_1 & -I & 0 & 0 & 0 & \dots & & 0 & 0 \\ C_1 & D_1 & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & A_1 & B_1 & -I & 0 & & & & \\ 0 & 0 & C_1 & D1 & 0 & 0 & & & & \\ \vdots & & & & & \ddots & & & & \\ & & & \dots & & & A_1 & B_1 & -I & 0 \\ & & & \dots & & & C_1 & D_1 & 0 & 0 \\ & & & \dots & & & 0 & 0 & C_1 & D1 \end{pmatrix}.$$

Then, both matrices have the same rank. □

Corollary 3.2.3. *Let (A, B, C, D) and (A_1, B_1, C_1, D_1) be two equivalent systems under equivalence relation considered. Then*

$$\operatorname{rank} T_\ell(A, B, C, D) = \operatorname{rank} T_\ell(A_1, B_1, C_1, D_1), \tag{3.11}$$

for all $\ell \in \mathbb{N}$.

Test for output-observability

Remark 6. It is obvious that if the matrix T_ℓ (consequently M_ℓ) has full row rank for some $\ell \in \mathbb{N}$, then all matrices T_j (consequently M_j) with $j \leq \ell$ have full row rank.

Moreover we have the following.

Proposition 3.2.3. *Let (A, B, C, D) be a system. For all $\ell \geq \delta$ we have*

$$\operatorname{rank} T_{\ell+1} - \operatorname{rank} T_\ell = \operatorname{rank} T_{\ell+2} - \operatorname{rank} T_{\ell+1} \tag{3.12}$$

Proof. Let (A, B, C, D) be a system, taking into account Proposition 3.2.2 and Corollary 3.2.3, we can consider an equivalent system in the form (A_1, B_1, C_1, D_1) with

$$A_1 = \begin{pmatrix} 0 & \bar{A}_1 \\ 0 & \bar{A}_2 \end{pmatrix},\ B_1 = \begin{pmatrix} \bar{B}_1 & 0 \\ \bar{B}_2 & 0 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} I_c & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},\ D_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & I_d \end{pmatrix}$$

with $\bar{A}_2 \in M_{\delta - c}(\mathbb{F})$.

In the case where $d = p$ the matrix $C_1 = 0$ and the result is obvious.

In the case where $d = k$ the matrix $B_1 = 0$. Calling $\bar{C} = \begin{pmatrix} I_c & 0 \\ 0 & 0 \end{pmatrix}$ and taking into account that $A^\delta = \sum_{i=0}^{\delta-1} A^i$, we get

$$\operatorname{rank} \begin{pmatrix} \bar{C} & & & & & & \\ 0 & I_d & & & & & \\ \bar{C}A & 0 & 0 & & & & \\ 0 & 0 & I_d & & & & \\ \vdots & & \ddots & & & & \\ \bar{C}A^{\delta-1} & 0 & 0 & 0 & \ddots & 0 & \\ 0 & 0 & 0 & 0 & \ddots & I_d & \\ 0 & 0 & 0 & 0 & \ddots & & 0 \\ 0 & 0 & 0 & 0 & \ddots & & I_d \\ \vdots & & & & & & \end{pmatrix} = d.$$

Then,

$$\operatorname{rank} T_{\ell+1} - \operatorname{rank} T_\ell = d,\ \textit{for all}\ \ell \geq \delta.$$

Suppose now, $d \neq p, k$. Firstly, we analyze the particular case where $\bar{A}_1 = 0$. It is easy to observe that

$$\operatorname{rank} T_{\ell+1} - \operatorname{rank}\ T_\ell = \operatorname{rank} \bar{B}_1 + d.$$

We observe that this case includes one of the more particular cases where $c = \delta$ and then $A_1 = 0$.

Then, we analyze the case $\bar{A}_1 \neq 0$. We have

$$\operatorname{rank} M_\ell = \operatorname{rank} \begin{pmatrix} I_r & & & & & & & & \\ & I_d & & & & & & & \\ & & \ddots & & & & & & \\ & & & I_d & & & & & \\ & & & & \bar{A}_1 & \bar{B}_1 & & & \\ & & & & \bar{A}_1\bar{A}_2 & \bar{A}_1\bar{B}_2 & \bar{B}_1 & & \\ & & & & \vdots & & & \ddots & \\ & & & & \bar{A}_1\bar{A}_2^{\ell} & \bar{A}_1\bar{A}_2^{\ell-1} & \dots & \bar{A}_1\bar{B}_2 & \bar{B}_1 \end{pmatrix}$$

Now, we consider the following reduced order system $(\bar{A}_2, \bar{B}_2, \bar{A}_1, \bar{B}_1)$ and we apply the previous steps. □

Corollary 3.2.4. *Let (A, B, C, D) be a system. For all $\ell \geq \delta$ we have:*

$$\operatorname{rank} M_{\ell+1} - \operatorname{rank} M_\ell = \operatorname{rank} M_{\ell+2} - \operatorname{rank} M_{\ell+1}.$$

As a Corollary, and taking into account Remark 6, we can conclude the following result.

Theorem 3.2.3 (Main Theorem)**.** *A system (A, B, C, D) is output observable if and only if the matrix M_δ has full row rank.*

This theorem provides an iterative method to compute output-observability.

Algorithm

Compute rank M_0
if rank$M_0 < p + \delta$ **then**
    the system is not output observable
else if rank $= p + \delta$ **then**
    Compute rank M_ℓ
end if
if rank $M_\ell < (\ell + 1)p + (\ell + 1)\delta$ **then**
    the system is not output observable
else if rank $= (\ell + 1)p + (\ell + 1)\delta$ and $\ell = \delta$ **then**
    the system is output observable

11: **else if** $\ell < \delta$ **then**
12: Go to step 5
13: **end if**

Example 3.2.2. In $\mathbb{F} = \mathbb{Z}_2$, we consider the (A, B, C, D) a representation of a convolutional code with $A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $C = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ and $D = 0_{3\times 1}$.

$$\operatorname{rank} \begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = 3$$

$$\operatorname{rank} \begin{pmatrix} C & D & 0 \\ CA & CB & D \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = 4 < 6.$$

Then, the system is not output observable.

3.3 Controllability of concatenated codes

We try to characterize the controllability of concatenated codes, from the property of the initial codes.

3.3.1 Serial concatenation

Let (A, B, C, D) be the serial concatenated code of the codes (A_i, B_i, C_i, D_i), $i = 1, 2$, as defined in (2.6).

The controllability character for serial concatenated codes can be described using the Hautus test in the following manner.

Theorem 3.3.1. *A serial concatenated code is controllable if and only if*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1 \\ -B_2C_1 & zI_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} = \delta_1 + \delta_2 \ \forall z \in \overline{\mathbb{F}}$$

From there, it is possible to deduce conditions in terms of both systems.

Corollary 3.3.1. *A necessary condition for controllability of serial concatenated code is that the pair (A_1, B_1) be controllable.*

Corollary 3.3.2. *A necessary condition for controllability of serial concatenated code is that the pair $(A_2, \overline{B})$ where $\overline{B} = \begin{pmatrix} -B_2C_1 & B_2D_1 \end{pmatrix}$ be controllable.*

Corollary 3.3.3. *A necessary condition for controllability of serial concatenated code is that the pair (A_2, B_2) be controllable.*

Proof.

$$\begin{aligned}\operatorname{rank} \begin{pmatrix} zI_2 - A_2 & B_2C_1 & B_2D_1 \end{pmatrix} &= \operatorname{rank} \begin{pmatrix} zI_2 - A_2 & B_2 & B_2 \end{pmatrix} \begin{pmatrix} I_2 & & \\ & C_1 & \\ & & D_1 \end{pmatrix} \\ &\leq \min \left(\operatorname{rank} \begin{pmatrix} zI_2 - A_2 & B_2 \end{pmatrix}, \operatorname{rank} \begin{pmatrix} I_2 & & \\ & C_1 & \\ & & D_1 \end{pmatrix} \right).\end{aligned}$$

So,

$$\operatorname{rank} \begin{pmatrix} zI_2 - A_2 & B_2C_1 & B_2D_1 \end{pmatrix} \leq \operatorname{rank} \begin{pmatrix} zI_2 - A_2 & B_2 \end{pmatrix} \leq \delta_2.$$

□

Example 3.3.1. Let (A, B, C, D) be a serial concatenated code of (A_1, B_1, C_1, D_1), and (A_2, B_2, C_2, D_2), where

$$A_1 = (0),\ B_1 = \begin{pmatrix} 1 & -3 \end{pmatrix},\ C_1 = (4),\ D_1 = \begin{pmatrix} 1 & -2 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},\ B_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ C_2 = \begin{pmatrix} 1 & 0 \end{pmatrix},\ D_2 = (1),$$

the concatenated serial code (A, B, C, D) of both is

$$A = \begin{pmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix} = \begin{pmatrix} 1 & -3 \\ 1 & -2 \\ 0 & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} D_2C_1 & C_2 \end{pmatrix} = \begin{pmatrix} 4 & 1 & 0 \end{pmatrix},\ D = \begin{pmatrix} D_2D_1 \end{pmatrix} = \begin{pmatrix} 1 & -2 \end{pmatrix}.$$

In this case is it easy to observe that all systems are controllable, this can be easily observed from the Hautus representation of our serial controllability matrix,

$$\begin{aligned}
&\operatorname{rank}\begin{pmatrix} zI_1 - A_1 & B_1 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} z & 1 & -3 \end{pmatrix} = 1, \text{ for all } z \in \overline{\mathbb{F}} \\
&\operatorname{rank}\begin{pmatrix} zI_2 - A_2 & B_2 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} z & 1 & 1 \\ -1 & z & 0 \end{pmatrix} = 2, \text{ for all } z \in \overline{\mathbb{F}} \\
&\operatorname{rank}\begin{pmatrix} zI_1 - A_1 & 0 & B_1 \\ -B_2C_1 & zI_2 - A_2 & B_2D_1 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} z & 0 & 0 & 1 & -3 \\ 0 & z & 1 & 1 & -2 \\ 0 & -1 & z & 0 & 0 \end{pmatrix} = 3,
\end{aligned}$$

for all $z \in \overline{\mathbb{F}}$.

Nevertheless the corollary only gives us a necessary condition, not sufficient, as we can see in the following example.

Example 3.3.2. Let (A, B, C, D) be a realization of the serial concatenated code of (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) where $A_1 = (1)$, $B_1 = (1)$, $C_1 = (1)$, $D_1 = (1)$, and $A_2 = (0)$, $B_2 = (1)$, $C_2 = (1)$ and $D_2 = (1)$.

Both systems are controllable because

$$\begin{aligned}
&\operatorname{rank}\begin{pmatrix} z - 1 & 1 \end{pmatrix} = 1 \text{ for all } z \in \overline{\mathbb{F}} \\
&\operatorname{rank}\begin{pmatrix} z & 1 \end{pmatrix} = 1 \text{ for all } z \in \overline{\mathbb{F}},
\end{aligned}$$

but the serial concatenated system (A, B, C, D) where

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 1 \end{pmatrix}, \; D = (1).$$

is not controllable because of

$$\operatorname{rank}\begin{pmatrix} zI - A & B \end{pmatrix} = \begin{pmatrix} z - 1 & 0 & 1 \\ -1 & z & 1 \end{pmatrix} = \begin{cases} 2 & \text{for all } z \neq 0, \\ 1 & \text{for } z = 0. \end{cases}$$

Remark 7. Obviously, if the matrix

$$\begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix}$$

has full row rank, then the concatenated serial system (A,B,C,D) is controllable.

A sufficient condition is obtained in the case where $Spec(A_1) \cap Spec(A_2) = \emptyset$.

Proposition 3.3.1. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the encoders $G_1(z)$ and $G_2(z)$ respectively with $Spec(A_1) \cap Spec(A_2) = \emptyset$. If the pairs (A_1, B_1) and (A_2, B_2) are controllable and the transfer matrix of (A_1, B_1, C_1, D_1) has full row rank for all $z \notin Spec(A_1)$, then the serial concatenated system is controllable.*

Proof. For all $z \notin Spec(A_1) \cup Spec(A_2)$, we have that

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & \\ -B_2C_1 & zI_{\delta_2} - A_2 \end{pmatrix} = \delta_1 + \delta_2,$$

so

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & & B_1 \\ -B_2C_1 & zI_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} = \delta_1 + \delta_2.$$

If $z_0 \in Spec(A_1)$, taking into account that $Spec(A_1) \cap Spec(A_2) = \emptyset$ we have that $\operatorname{rank}(z_0I_{\delta_2} - A_2) = \delta_2$, then

$$\begin{aligned} &\operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & & B_1 \\ -B_2C_1 & z_0I_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} \\ &= \operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & B_1 & \\ -B_2C_1 & B_2D_1 & z_0I_{\delta_2} - A_2 \end{pmatrix} \\ &= \operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & B_1 \end{pmatrix} + \delta_2. \end{aligned}$$

But, taking into account that (A_1, B_1) is controllable we have $\operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & B_1 \end{pmatrix} = \delta_1$. Then,

$$\operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & & B_1 \\ -B_2C_1 & z_0I_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} = \delta_1 + \delta_2.$$

Finally, if $z_0 \in Spec(A_2)$, we have that $\operatorname{rank}(z_0I_{\delta_1} - A_1) = \delta_1$, then

$$\begin{aligned} &\operatorname{rank} \begin{pmatrix} z_0I_{\delta_1} - A_1 & & B_1 \\ -B_2C_1 & z_0I_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} \\ &= \operatorname{rank} \begin{pmatrix} I_{\delta_1} & 0 & \\ 0 & z_0I_{\delta_2} - A_2 & B_2C_1(z_0I - A_1)^{-1}B_1 + B_2D_1 \end{pmatrix} \\ &= \delta_1 + \operatorname{rank} \begin{pmatrix} z_0I - A_2 & B_2 \end{pmatrix} \begin{pmatrix} I_{\delta_2} & 0 \\ & C_1(z_0I - A_1)^{-1}B_1 + D_1 \end{pmatrix} \\ &= \delta_1 + \delta_2, \text{for all } z_0 \in Spec(A_2), \end{aligned}$$

knowing that (A_2, B_2) is controllable and $C_1(z_0 I - A_1)^{-1} B_1 + D_1$ has full row rank. □

Remark 8. Example 3.3.2 shows that if $C_1(z_0 I - A_1)^{-1} B_1 + D_1$ does no t have full rank for all $z_0 \in Spec(A_2)$ the Result 3.3.1 is not true.

3.3.2 Parallel concatenation

In the parallel concatenation case, the same entries are used for both codes. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be convolutional codes.

The input information $u_2(t) = u_1(t) = u(t)$ and the final parity vector $y(t) = y_1(t) + y_2(t)$.

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}, \; B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}$$
$$C = \begin{pmatrix} C_1 & C_2 \end{pmatrix}, \; D = D_1 + D_2.$$

If $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ is a (n, k, δ_1)-code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ is a (n, k, δ_2)-code, then $\mathcal{C}(A, B, C, D)$ is a $(n, k, \delta_1 + \delta_2)$-code.

So, we have the following result.

Theorem 3.3.2. *The parallel concatenated code $\mathcal{C}(A, B, C, D)$ is controllable, if and only if the following matrix*

$$\begin{pmatrix} B_1 & A_1 B_1 & \dots & A_1^{\delta_1+\delta_2-1} B_1 \\ B_2 & A_2 B_2 & \dots & A_2^{\delta_1+\delta_2-1} B_2 \end{pmatrix}$$

has full rank.

Or, using the Hautus test, we have

Theorem 3.3.3. *The parallel concatenated code $\mathcal{C}(A, B, C, D)$ is controllable, if and only if the following matrix*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & & B_1 \\ & zI_{\delta_2} - A_2 & B_2 \end{pmatrix} = \delta_1 + \delta_2, \quad \textit{for all } z \in \overline{\mathbb{F}}.$$

Proposition 3.3.2. *A necessary condition for controllability of parallel concatenated system is that the pairs (A_1, B_1) and (A_2, B_2) are controllable.*

Unfortunately, this condition is only necessary, but not sufficient. As we can see in this particular case:

Example 3.3.3. For instance, let us have two realisations:

$$A_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, B_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, D_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Indeed, (A_1, B_1) is controllable,

$$\operatorname{rank} \begin{pmatrix} B_1 & A_1 B_1 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = 2$$

$$A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, D_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

(A_2, B_2) is controllable as well,

$$\operatorname{rank} \begin{pmatrix} B_2 & A_2 B_2 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = 2$$

However, the parallel concatenated model

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \ C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \ D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

is not controllable since

$$\operatorname{rank} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 2 < 4.$$

A sufficient condition is obtained in the case where $Spec(A_1) \cap Spec(A_2) = \emptyset$.

Proposition 3.3.3. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the encoders $G_1(z)$ and $G_2(z)$ respectively with $Spec(A_1) \cap Spec(A_2) = \emptyset$. If the pairs (A_1, B_1) and (A_2, B_2) are controllable, then the parallel concatenated system is controllable.*

Proof. For all $z \notin Spec(A_1) \cup Spec(A_2)$, we have that

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & \\ & zI_{\delta_2} - A_2 \end{pmatrix} = \delta_1 + \delta_2,$$

so

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & & B_1 \\ & zI_{\delta_2} - A_2 & B_2 \end{pmatrix} = \delta_1 + \delta_2.$$

If $z_0 \in Spec(A_1)$, taking into account that $Spec(A_1) \cap Spec(A_2) = \emptyset$ we have that $\operatorname{rank}(z_0 I_{\delta_2} - A_2) = \delta_2$, then

$$\begin{aligned} \operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & & B_1 \\ & zI_{\delta_2} - A_2 & B_2 \end{pmatrix} &= \operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & B_1 & \\ & B_2 & zI_{\delta_2} - A_2 \end{pmatrix} \\ &= \operatorname{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & B_1 \end{pmatrix} + \delta_2. \end{aligned}$$

But, taking into account that (A_1, B_1) is controllable we have $\operatorname{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & B_1 \end{pmatrix} = \delta_1$. Then,

$$\operatorname{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & & B_1 \\ & z_0 I_{\delta_2} - A_2 & B_2 \end{pmatrix} = \delta_1 + \delta_2.$$

Analogously, if $z_0 \in Spec(A_2)$, we have that $\operatorname{rank}(z_0 I_{\delta_1} - A_1) = \delta_1$, then

$$\operatorname{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & & B_1 \\ & z_0 I_{\delta_2} - A_2 & B_2 \end{pmatrix} = \delta_1 + \delta_2, \text{for all } z_0 \in Spec(A_2).$$

□

Remark 9. Example 3.3.3 shows that if $Spec(A_1) \cap Spec(A_2) \neq \emptyset$ the result 3.3.3 is not true.

3.3.3 Systematic serial concatenation

Let (A, B, C, D) be the systematic serial concatenated code of (A_i, B_i, C_i, D_i), $i = 1, 2$, codes defined in (2.6).

As in the two previous cases, the controllability character for systematic serial concatenated codes can be described in terms of the both systems using the Hautus test in the following manner.

Theorem 3.3.4. *A systematic serial controllability concatenated code of* (A_i, B_i, C_i, D_i), $i = 1, 2$ *is controllable, if and only if.*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1 \\ -B_2C_1 & zI_{\delta_2} - A_2 & B_2D_1 \end{pmatrix} = \delta_1 + \delta_2 \ \forall z \in \overline{\mathbb{F}}$$

We observe that this result coincides with the case of serial concatenation. Then all results about controllability of serial concatenation are valid for systematic serial concatenation and vice-versa.

3.3.4 Parallel interleaver concatenation

Considering the parallel interleaver concatenated code of (A_i, B_i, C_i, D_i), $i = 1, 2$, with the interleaver matrix $\mathcal{P}$, the parallel interleaver controllability concatenated condition can be obtained from the Hautus test:

Theorem 3.3.5. *The parallel interleaver concatenated code* $\mathcal{C}(A, B, C, D)$ *is controllable if and only if*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1 \\ 0 & zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \delta_1 + \delta_2 \ \forall z \in \overline{\mathbb{F}}.$$

From this theorem, we obtain some necessary and sufficient conditions for observability of this concatenated code.

Proposition 3.3.4. *A necessary condition for controllability of the parallel interleaver concatenated code of* (A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2)*, with the interleaver matrix* $\mathcal{P}$ *is that the pair* (A_1, B_1) *be controllable.*

Proposition 3.3.5. *A necessary condition for controllability of the parallel interleaver concatenated code of* (A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2)*, with the interleaver matrix* $\mathcal{P}$ *is that the pair* (A_2, B_2) *be controllable.*

Proof. In case

$$\begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1 \\ 0 & zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix}$$

has full row rank. Then the block $\begin{pmatrix} 0 & zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix}$ has full row rank.

Moreover, we can observe that:

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \operatorname{rank}\begin{pmatrix} zI_{\delta_2} - A_2 & B_2 \end{pmatrix}\begin{pmatrix} I_{\delta_2} & \\ & \mathcal{P} \end{pmatrix}.$$

□

Proposition 3.3.6. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the encoders $G_1(z)$ and $G_2(z)$ respectively, and $\mathcal{P}$ the matrix for interleaving. Then, the concatenated system of both codes is controllable under the parallel interleaver model if and only if the concatenated code of $(A_1, B_1\mathcal{P}^{-1}, C_1, D_1)$ and (A_2, B_2, C_2, D_2) is controllable under the parallel model.*

Proof. When trying to compute the controllability character, we realize that:

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1 \\ 0 & zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} =$$

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & 0 & B_1\mathcal{P}^{-1} \\ 0 & zI_{\delta_2} - A_2 & B_2 \end{pmatrix}\begin{pmatrix} I_{\delta_1} & & \\ & I_{\delta_2} & \\ & & \mathcal{P} \end{pmatrix}.$$

□

A sufficient condition is obtained in the case where $Spec(A_1) \cap Spec(A_2) = \emptyset$.

Proposition 3.3.7. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the encoders $G_1(z)$ and $G_2(z)$ respectively, and $\mathcal{P}$ the matrix for interleaving, with $Spec(A_1) \cap Spec(A_2) = \emptyset$. If the pairs (A_1, B_1) and (A_2, B_2) are controllable, then the parallel interleaver concatenated system is controllable.*

Proof. For all $z \notin Spec(A_1) \cup Spec(A_2)$, we have that

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & \\ & zI_{\delta_2} - A_2 \end{pmatrix} = \delta_1 + \delta_2,$$

so

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & & B_1 \\ & zI_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \delta_1 + \delta_2.$$

If $z_0 \in Spec(A_1)$, taking into account that $Spec(A_1) \cap Spec(A_2) = \emptyset$ we have that $\text{rank}\,(z_0 I_{\delta_2} - A_2) = \delta_2$, then

$$\begin{aligned}\text{rank}\begin{pmatrix} z_0 I_{\delta_1} - A_1 & & B_1 \\ & z_0 I_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} &= \text{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & B_1 & \\ & B_2\mathcal{P} & zI_{\delta_2} - A_2 \end{pmatrix} \\ &= \text{rank}\begin{pmatrix} z_0 I_{\delta_1} - A_1 & B_1 \end{pmatrix} + \delta_2.\end{aligned}$$

But, taking into account that (A_1, B_1) is controllable we have $\text{rank}\begin{pmatrix} z_0 I_{\delta_1} - A_1 & B_1 \end{pmatrix} = \delta_1$. Then,

$$\text{rank}\begin{pmatrix} z_0 I_{\delta_1} - A_1 & & B_1 \\ & z_0 I_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \delta_1 + \delta_2.$$

If $z_0 \in Spec(A_2)$, we have that $\text{rank}\,(z_0 I_{\delta_1} - A_1) = \delta_1$, then

$$\text{rank}\begin{pmatrix} z_0 I_{\delta_1} - A_1 & & B_1 \\ & z_0 I_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \delta_1 + \text{rank}\begin{pmatrix} z_0 I_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} =$$
$$\delta_1 + \text{rank}\begin{pmatrix} z_0 I_{\delta_2} - A_2 & B_2 \end{pmatrix}\begin{pmatrix} I_{\delta_2} & \\ & \mathcal{P} \end{pmatrix} = \delta_1 + \delta_2, \text{for all } z_0 \in Spec(A_2),$$

knowing that $\text{rank}\begin{pmatrix} z_0 I_{\delta_2} - A_2 & B_2\mathcal{P} \end{pmatrix} = \text{rank}\begin{pmatrix} z_0 I_{\delta_2} - A_2 & B_2 \end{pmatrix}\begin{pmatrix} I_{\delta_2} & \\ & \mathcal{P} \end{pmatrix}$. □

3.4 Observability of concatenated codes

3.4.1 Serial concatenation

The serial observability concatenated character is obtained from the Hautus test:

Theorem 3.4.1. *The serial concatenated code $\mathcal{C}(A, B, C, D)$ of (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) is observable if and only if:*

$$\text{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \\ D_2C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2 \ \forall z \in \overline{\mathbb{F}}.$$

Corollary 3.4.1. *A necessary condition for observability of concatenated code is that the pair (A_2, C_2) be observable.*

Corollary 3.4.2. *A necessary condition for observability of concatenated code is that the pair* $(A_1, \bar{C_1})$*, with* $\bar{C_1} = \begin{pmatrix} -B_2C_1 \\ D_2C_1 \end{pmatrix}$ *be observable.*

Corollary 3.4.3. *A necessary condition for observability of concatenated code is that the pair* (A_1, C_1)*, be observable.*

Proof.

$$\begin{aligned} \operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 \\ -B_2C_1 \\ D_2C_1 \end{pmatrix} &= \operatorname{rank} \begin{pmatrix} I_{\delta_1} & & \\ & -B_2 & \\ & & D_2 \end{pmatrix} \begin{pmatrix} zI_{\delta_1} - A_1 \\ C_1 \\ C_1 \end{pmatrix} \\ &\leq \min \left(\operatorname{rank} \begin{pmatrix} I_{\delta_1} & & \\ & -B_2 & \\ & & D_2 \end{pmatrix}, \operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 \\ C_1 \\ C_1 \end{pmatrix} \right) \end{aligned}$$

Then

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 \\ -B_2C_1 \\ D_2C_1 \end{pmatrix} \leq \operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 \\ C_1 \end{pmatrix} \leq \delta_1.$$

□

In this specific case, a sufficient condition is obtained after observation of the spectrum of both matrices A_1 and A_2.

Proposition 3.4.1. *Let* (A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2) *be realizations of the encoders* $G_1(z)$ *and* $G_2(z)$ *respectively with* $Spec(A_1) \cap Spec(A_2) = \emptyset$*. If the pairs* (A_1, C_1) *and* (A_2, C_2) *are observable, with* B_2 *has full column rank, then the serial concatenated system is observable.*

Proof. Let us suppose that we have: $z \notin Spec(A_1) \cup Spec(A_2)$; then,

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \end{pmatrix} = \delta_1 + \delta_2, \forall z \in \overline{\mathbb{F}}$$

which means that:

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \\ D_2C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2, \forall z \in \overline{\mathbb{F}}$$

Let us consider $z_0 \in Spec(A_2)$; knowing that $Spec(A_1) \cap Spec(A_2) = \emptyset$, then rank $\left(z_0 I_{\delta_1} - A_1\right) = \delta_1$

then,

$$\text{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & 0 \\ -B_2 C_1 & z_0 I_{\delta_2} - A_2 \\ D_2 C_1 & C_2 \end{pmatrix} = \delta_1 + \text{rank} \begin{pmatrix} z_0 I_{\delta_2} - A_2 \\ C_2 \end{pmatrix} = \delta_1 + \delta_2$$

since (A_2, C_2) is observable.

Let us consider $z_0 \in Spec(A_1)$; knowing that $Spec(A_1) \cap Spec(A_2) = \emptyset$, then rank $\left(z_0 I_{\delta_2} - A_2\right) = \delta_2$

$$\text{then, rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 & 0 \\ -B_2 C_1 & z_0 I_{\delta_2} - A_2 \\ D_2 C_1 & C_2 \end{pmatrix} = \delta_2 + \text{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 \\ -B_2 C_1 \\ D_2 C_1 \end{pmatrix}$$

If we suppose that B_2 has full column rank, we can see that:

$$\text{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 \\ -B_2 C_1 \\ D_2 C_1 \end{pmatrix} = \text{rank} \begin{pmatrix} I_{\delta_1} & & \\ & -B_2 & \\ & & D_2 \end{pmatrix} \begin{pmatrix} z_0 I_{\delta_1} - A_1 \\ C_1 \\ C_1 \end{pmatrix}$$
$$= \text{rank} \begin{pmatrix} z_0 I_{\delta_1} - A_1 \\ C_1 \\ 0 \\ D_2 C_1 \end{pmatrix} = \delta_1$$

since (A_1, C_1) is observable

Which means that:

$$\text{rank} \begin{pmatrix} z I_{\delta_1} - A_1 & 0 \\ -B_2 C_1 & z I_{\delta_2} - A_2 \\ D_2 C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2, \text{for all} z \in \overline{\mathbb{F}} \text{ when } Spec(A_1) \cap Spec(A_2) = \emptyset$$

□

3.4.2 Parallel concatenation

In parallel concatenated model $\mathcal{C}(A, B, C, D)$ obtained from the convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, the observability matrix is

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\delta_1+\delta_2-1} \end{pmatrix} = \begin{pmatrix} C_1 & C_2 \\ C_1A_1 & C_2A_2 \\ C_1A_1^2 & C_2A_2^2 \\ \vdots & \\ C_1A_1^{\delta_1+\delta_2-1} & C_2A_2^{\delta_1+\delta_2-1} \end{pmatrix} \tag{3.13}$$

So, we have the following theorem.

Theorem 3.4.2. *The parallel concatenated code $\mathcal{C}(A, B, C, D)$ is observable, if and only if, the matrix* (3.13)*, has full rank.*

Using the Hautus test we have

Theorem 3.4.3. *The parallel concatenated code $\mathcal{C}(A, B, C, D)$ is observable, if and only if the following matrix*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & \\ & zI_{\delta_2} - A_2 \\ C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2, \ \forall z \in \overline{\mathbb{F}}.$$

Proposition 3.4.2. *A necessary condition for observability of parallel concatenated system is that the pairs (A_1, C_1) and (A_2, C_2) are observable.*

Nevertheless, this condition is not sufficient as we can see in the following example.

Example 3.4.1. Let $\mathcal{C}(A, B, C, D)$ be the parallel concatenated code of the following realizations. The first realization is defined by the system (A_1, B_1, C_1, D_1) with $A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$, $B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$, $C_1 = \begin{pmatrix} 2 & 1 & 0 \end{pmatrix}$ and $D_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}$,

and the second realization is defined by the system (A_2, B_2, C_2, D_2) with

$$A_2 = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}, \ B_2 = \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \ C_2 = \begin{pmatrix} 2 & 0 & 1 \end{pmatrix}, \ D_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

Both codes are observable since

$$\operatorname{rank}\begin{pmatrix} C_1 \\ C_1A_1 \\ C_1A_1^2 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{pmatrix} = 3,$$

$$\operatorname{rank}\begin{pmatrix} C_2 \\ C_2A_2 \\ C_2A_2^2 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 2 & 0 & 1 \\ -1 & -2 & 0 \\ 3 & 1 & 2 \end{pmatrix} = 3.$$

However, the parallel concatenated model is not observable since

$$\operatorname{rank}\begin{pmatrix} 2 & 1 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & -1 & -2 & 0 \\ 1 & 0 & 2 & 3 & 1 & 2 \\ 2 & 1 & 0 & -2 & -3 & -1 \\ 0 & 2 & 1 & 4 & 2 & 3 \\ 1 & 0 & 2 & -3 & -4 & -2 \end{pmatrix} = 5 < 6.$$

Proposition 3.4.3. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the encoders $G_1(z)$ and $G_2(z)$ respectively with $Spec(A_1) \cap Spec(A_2) = \emptyset$. If the pairs (A_1, C_1) and (A_2, C_2) are observable, then the parallel concatenated system is observable.*

Proof. Analogous to 3.3.3. □

3.4.3 Systematic serial concatenation

In the systematic serial concatenated code $\mathcal{C}(A, B, C, D)$ obtained from the convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, the observability concatenated character is obtained from the Hautus test:

Theorem 3.4.4. *The systematic serial concatenated code $\mathcal{C}(A, B, C, D)$ is observable if and only if the following relation holds.*

$$\operatorname{rank}\begin{pmatrix} zI_{\delta_1} - A_1 & 0 \\ -B_2C_1 & zI_{\delta_2} - A_2 \\ C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2 \text{ for all} z \in \overline{\mathbb{F}}$$

After this theorem it is obvious the following sufficient condition for observability of systematic serial concatenated code.

Corollary 3.4.4. *A sufficient condition for observability of systematic serial concatenated code is that the serial concatenated code is.*

Proof.

$$\delta_1+\delta_2=\text{rank}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ -B_2C_1 & zI_{\delta_2}-A_2 \\ D_2C_1 & C_2 \end{pmatrix} \leq \text{rank}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ -B_2C_1 & zI_{\delta_2}-A_2 \\ C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix} \leq \delta_1+\delta_2.$$

□

In this case we have a necessary and sufficient condition depending only on the conditions of the initial codes.

Theorem 3.4.5. *A necessary and sufficient condition for observability of systematic serial concatenated code is that both codes are observable.*

Proof.

$$\begin{aligned} &\text{rank}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ -B_2C1 & zI_{\delta_2}-A_2 \\ C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix} \\ &= \text{rank}\begin{pmatrix} I_{\delta_1} & & \\ & I_{\delta_2} & B_2 \\ & & I_{m-k} \\ & & -D_2 \quad I_{n-m+k} \end{pmatrix}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ -B_2C1 & zI_{\delta_2}-A_2 \\ C_1 & 0 \\ D_2C_1 & C_2 \end{pmatrix} \\ &= \text{rank}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ 0 & zI_{\delta_2}-A_2 \\ C_1 & 0 \\ 0 & C_2 \end{pmatrix} = \text{rank}\begin{pmatrix} zI_{\delta_1}-A_1 & 0 \\ C_1 & 0 \\ 0 & zI_{\delta_2}-A_2 \\ 0 & C_2 \end{pmatrix}. \end{aligned}$$

□

Example 3.4.2. In $\mathbb{F}_5$, we consider the codes $\mathcal{C}_1(A_1,B_1,C_1,D_1)$, $\mathcal{C}_2(A_2,B_2,C_2,D_2)$ concatenated in a serial systematic form. The matrices defining the codes are

$$A_1=\begin{pmatrix}1&3&1\\0&2&4\\3&0&2\end{pmatrix},\ B_1=\begin{pmatrix}0\\1\\2\end{pmatrix},\ C_1=\begin{pmatrix}1&0&2\end{pmatrix},\ D_1=\begin{pmatrix}3\end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 1 & 3 & 0 \\ 2 & 1 & 2 \\ 1 & 0 & 4 \end{pmatrix},\ B_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix},\ C_2 = \begin{pmatrix} 1 & 4 & 2 \\ 1 & 0 & 4 \end{pmatrix},\ D_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}.$$

Taking into account that

$$\operatorname{rank} \begin{pmatrix} C_1 \\ C_1A_1 \\ C_1A_1^2 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 0 & 2 \\ 2 & 3 & 0 \\ 2 & 4 & 4 \end{pmatrix} = 3$$

$$\operatorname{rank} \begin{pmatrix} C_2 \\ C_2A_2 \\ C_2A_2^2 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 4 & 2 \\ 1 & 0 & 4 \\ 1 & 2 & 1 \\ 0 & 3 & 3 \\ 1 & 0 & 2 \\ 2 & 3 & 3 \end{pmatrix} = 3$$

both codes are observable.

Then, applying theorem 3.4.5, the concatenated code $\mathcal{C}$ is observable.

We can check the result:

$$A = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 \\ 2 & 0 & 4 & 2 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 & 4 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 0 \\ 1 \\ 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 4 & 2 \\ 1 & 0 & 2 & 1 & 0 & 4 \end{pmatrix},\ D = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

$$\text{and } \operatorname{rank} \begin{pmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 4 & 2 \\ 1 & 0 & 2 & 1 & 0 & 4 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 4 & 4 & 1 & 1 & 2 & 1 \\ 2 & 3 & 0 & 0 & 3 & 1 \\ 4 & 0 & 3 & 0 & 0 & 0 \\ 2 & 4 & 1 & 2 & 3 & 2 \\ 2 & 3 & 0 & 0 & 1 & 1 \end{pmatrix} = 6.$$

Then, the systematic serial concatenated matrix is observable.

3.4.4 Parallel interleaver concatenation

The observability condition for the parallel interleaver concatenated system of two convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, and interleaver matrix $\mathcal{P}$ is given by the following theorem

Theorem 3.4.6. *A parallel interleaver concatenated system is observable, if and only if*

$$\operatorname{rank} \begin{pmatrix} zI_{\delta_1} - A_1 & \\ & zI_{\delta_2} - A_2 \\ C_1 & C_2 \end{pmatrix} = \delta_1 + \delta_2, \ \forall z \in \overline{\mathbb{F}}.$$

Indeed, the observability character of this concatenated model is the same as the parallel concatenated model. In this concatenated model all results for parallel concatenation are the same for the observability property.

3.5 Output-observability of concatenated codes

3.5.1 Serial concatenation

Unlike the cases of controllability and observability of serial concatenated codes, the output-observability of the convolutional codes is not a necessary condition for output-observability of serial concatenated code obtained from these codes, as we can see in the following example.

Example 3.5.1. Over the field $\mathbb{F} = \mathbb{Z}_5$, we consider the following realizations (A_1, B_1, C_1, D_1), and (A_2, B_2, C_2, D_2) of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, with

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ B_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \ D_1 = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}$$

and

$$A_2 = A_1, \ B_2 = B_1, \ C_2 = C_1, \ D_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

The serial concatenated system considered is (A, B, C, D) with

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 4 & 0 \\ 0 & 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

In this case, the system (A_1, B_1, C_1, D_1) is output observable but neither (A_2, B_2, C_2, D_2) nor the serial concatenated system (A, B, C, D) are output observable:

$$\operatorname{rank} \begin{pmatrix} C_1 & D_1 & & \\ C_1A_1 & C_1B_1 & D_1 & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 \end{pmatrix} =$$

$$\operatorname{rank} \begin{pmatrix} 1 & 0 & 1 & 0 & & & & \\ 0 & 0 & -1 & 0 & & & & \\ 0 & 1 & 0 & 1 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & -1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} = 6.$$

$$\operatorname{rank} \begin{pmatrix} C_2 & D_2 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 1$$

And

$$\operatorname{rank} \begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = 1.$$

But sometimes it is possible that the three codes are output-observable.

Example 3.5.2. Over the field $\mathbb{F} = \mathbb{Z}_5$, we consider the realizations (A_1, B_1, C_1, D_1), and (A_2, B_2, C_2, D_2) of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, with

$$A_1 = \begin{pmatrix} 0 \end{pmatrix},\ B = \begin{pmatrix} 1 & 2 \end{pmatrix},\ C = \begin{pmatrix} 4 \end{pmatrix},\ D = \begin{pmatrix} 1 & 3 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix},\ B_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ C_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}, D_2 = \begin{pmatrix} 1 \end{pmatrix}$$

The serial concatenated code considered is (A, B, C, D) with

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & 4 \\ 0 & 1 & 0 \end{pmatrix},\ B = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 0 & 0 \end{pmatrix},\ C = \begin{pmatrix} 4 & 1 & 0 \end{pmatrix},\ D = \begin{pmatrix} 1 & 3 \end{pmatrix}$$

It is easy to show that all systems (A_1, B_1, C_1, D_1), (A_2, B_2, C_2, D_2) and (A, B, C, D) are output observable:

$$\operatorname{rank}\begin{pmatrix} C_1 & D_1 & \\ C_1A_1 & C_1B_1 & D_1 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 4 & 1 & 3 & & \\ 0 & 4 & 3 & 1 & 3 \end{pmatrix} = 2$$

$$\operatorname{rank}\begin{pmatrix} C_2 & D_2 & & \\ C_2A_2 & C_2B_2 & D_2 & \\ C_2A_2^2 & C_2A_2B_2 & C_2B_2 & D_2 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & 0 & 1 & & \\ 0 & 4 & 1 & 1 & \\ 4 & 0 & 0 & 1 & 1 \end{pmatrix} = 3$$

And

$$\operatorname{rank}\begin{pmatrix} C & D & & & \\ CA & CB & D & & \\ CA^2 & CAB & CB & D & \\ CA^3 & CA^2B & CAB & CB & D \end{pmatrix} =$$

$$\operatorname{rank}\begin{pmatrix} 4 & 1 & 0 & 1 & 3 & & & & & \\ 4 & 0 & 4 & 0 & 1 & 1 & 3 & & & \\ 0 & 4 & 0 & 4 & 3 & 0 & 1 & 1 & 3 & \\ 1 & 0 & 1 & 4 & 2 & 4 & 3 & 0 & 1 & 1 & 3 \end{pmatrix} = 4.$$

The output-observability character of each of the systems (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) it is not sufficient for output-observability character of the serial concatenated system (A, B, C, D).

Example 3.5.3. Over a finite field $\mathbb{F}$, we consider the realizations (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) of the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ respectively, with

$$A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},\ B_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},\ C_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},\ D_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and

$$A_2 = A_1,\ B_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},\ C_2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ D_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

The serial concatenated system considered is (A, B, C, D) with

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix},\ C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ D = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

In this case, the system (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) are output observable but the serial concatenated system (A, B, C, D) is not output observable:

$$\operatorname{rank}\begin{pmatrix} C_1 & D_1 & & \\ C_1A_1 & C_1B_1 & D_1 & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 0 & 0 & 0 & 1 & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = 6$$

$$\operatorname{rank}\begin{pmatrix} C_2 & D_2 & & \\ C_2A_2 & C_2B_2 & D_2 & \\ C_2A_2^2 & C_2A_2B_2 & C_2B_2 & D_2 \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = 6$$

And

$$\operatorname{rank}\begin{pmatrix} C & D & \\ CA & CB & D \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = 3 < 4.$$

We can try now to obtain conditions for output-observability of concatenated codes. The first one is a sufficient condition that follows from 3.2.2.

Corollary 3.5.1. *Let $\mathcal{C}(A, B, C, D)$ be the serial concatenated code obtained from the concatenation of the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$. If matrix D_2D_1 has full row rank, then the code $\mathcal{C}(A, B, C, D)$ is output observable.*

Unfortunately, as seen in the previous cases, when this case is not really satisfied, we can still observe a necessary condition for the output-observability in terms of the matrices C_1, C_2, D_1 and D_2, defining the codes.

Let T_0 be the first line matrix block code of our output controllability matrix T_ℓ.

$$T_0 = \begin{pmatrix} D_2C_1 & C_2 & D_2D_1 \end{pmatrix}.$$

Proposition 3.5.1. *A necessary condition for output-observability of the serial concatenated code $\mathcal{C}(A, B, C, D)$ is that the matrix $\begin{pmatrix} C_2 & D_2 \end{pmatrix}$ has full row rank.*

Proof. If the code $\mathcal{C}(A, B, C, D)$ is output-observable the matrix T_0 has full row rank. Then, taking as well into account that

$$\begin{aligned} &\operatorname{rank} \begin{pmatrix} D_2C_1 & C_2 & D_2D_1 \end{pmatrix} \\ &= \operatorname{rank} \begin{pmatrix} D_2 & C_2 & D_2 \end{pmatrix} \begin{pmatrix} C_1 & & \\ & I & \\ & & D_1 \end{pmatrix} \\ &\leq \min \left(\operatorname{rank} \begin{pmatrix} D_2 & C_2 & D_2 \end{pmatrix}, \operatorname{rank} \begin{pmatrix} C_1 & & \\ & I & \\ & & D_1 \end{pmatrix} \right) \end{aligned}$$

we have

$$\operatorname{rank} T_0 \leq \operatorname{rank} \begin{pmatrix} C_2 & D_2 \end{pmatrix}.$$

□

Let recall $GL_n(\mathbb{F})$ the group of all invertible $n \times n$ matrices. Suppose now $k = p$, we have the following proposition

Proposition 3.5.2. *Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ two convolutional codes with $(A_2, B_2, C_2, D_2) = (P^{-1}A_1P, P^{-1}B_1, C_1P, D_1)$ for some invertible matrix $P \in Gl(\delta_1, \mathbb{F})$. Then, the serial concatenated code $\mathcal{C}(A, B, C, D)$ of these codes is output observable, if and only if the concatenated convolutional code $\mathcal{C}(\overline{A}, \overline{B}, \overline{C}, \overline{D})$ of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ with itself is output observable.*

Proof. First of all we observe that the codes $\mathcal{C}_i(A_i, B_i, C_i, D_i)$ can be concatenated in a serial form because of $k = p$.

Now, it suffices to observe that

$$\operatorname{rank}\begin{pmatrix}
A_1 & 0 & B_1 & -I_1 & 0 & & & & & & & \\
B_2C_1 & A_2 & B_2D_1 & 0 & -I_1 & & & & & & & \\
D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & & \\
0 & 0 & 0 & B_2C_1 & A_2 & B_2D_1 & 0 & -I_1 & & & & \\
0 & 0 & 0 & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & & \\
 & & & & & & \ddots & \ddots & \ddots & & & \\
 & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 \\
 & & & 0 & 0 & 0 & B_2C_1 & A_2 & B_2D_1 & 0 & -I_1 & 0 \\
 & & & 0 & 0 & 0 & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & D_2C_1 & C_2 & D_2D_1
\end{pmatrix}$$

$$= \operatorname{rank}\mathbf{Q}\cdot\begin{pmatrix}
A_1 & 0 & B_1 & -I_1 & 0 & & & & & & & \\
B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & & & & & & & \\
D_1C_1 & C_1 & D_1^2 & 0 & 0 & & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & & \\
0 & 0 & 0 & B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & & & & \\
0 & 0 & 0 & D_1C_1 & C_1 & D_1^2 & 0 & 0 & & & & \\
 & & & & & & \ddots & \ddots & \ddots & & & \\
 & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 \\
 & & & 0 & 0 & 0 & B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & 0 \\
 & & & 0 & 0 & 0 & D_1C_1 & C_1 & D_1^2 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & D_1C_1 & C_1 & D_1^2
\end{pmatrix}\cdot\mathbf{P}$$

$$= \operatorname{rank}\begin{pmatrix}
A_1 & 0 & B_1 & -I_1 & 0 & & & & & & & \\
B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & & & & & & & \\
D_1C_1 & C_1 & D_1^2 & 0 & 0 & & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & & \\
0 & 0 & 0 & B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & & & & \\
0 & 0 & 0 & D_1C_1 & C_1 & D_1^2 & 0 & 0 & & & & \\
 & & & & & & \ddots & \ddots & \ddots & & & \\
 & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 \\
 & & & 0 & 0 & 0 & B_1C_1 & A_1 & B_1D_1 & 0 & -I_1 & 0 \\
 & & & 0 & 0 & 0 & D_1C_1 & C_1 & D_1^2 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & D_1C_1 & C_1 & D_1^2
\end{pmatrix}$$

Where the matrices $\mathbf{Q}$ and $\mathbf{P}$ are:

$$\mathbf{Q} = \begin{pmatrix}
I_1 & & & & & & & & & & & \\
 & P^{-1} & & & & & & & & & & \\
 & & I_2 & & & & & & & & & \\
 & & & I_1 & & & & & & & & \\
 & & & & \ddots & & & & & & & \\
 & & & & & P^{-1} & & & & & & \\
 & & & & & & I_2 & & & & & \\
 & & & & & & & I_1 & & & &
\end{pmatrix}$$

and

$$\mathbf{P} = \begin{pmatrix} I_1 & & & & & & \\ & P & & & & & \\ & & I_2 & & & & \\ & & & I_1 & & & \\ & & & & \ddots & & \\ & & & & & P & \\ & & & & & & I_2 \end{pmatrix}$$

with I_1 and I_2 the identity matrices in the adequate size. □

Example 3.5.4. Over $\mathbb{F}_4$, we consider the representation of a convolutional code (A_1, B_1, C_1, D_1) with $A_1 = (\alpha)$, $B_1 = (1)$, $C_1 = (\alpha + 1)$ and $D_1 = (1)$. The serial concatenated code of this code with itself is (A, B, C, D) with

$$A = \begin{pmatrix} \alpha & 0 \\ \alpha + 1 & \alpha \end{pmatrix}, \; B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; C = \begin{pmatrix} \alpha + 1 & \alpha + 1 \end{pmatrix}, \; D = (1)$$

is clearly output-observable, then after Proposition 3.5.2 the serial concatenated system of (A_1, B_1, C_1, D_1) with $(P^{-1}A_1P, P^{-1}B_1, C_1P, D_1)$ is output-observable for all $P \in Gl(1, \mathbb{F})$.

3.5.2 Parallel concatenation

Like the case of serial concatenation, the output-observability of the convolutional codes is not a necessary condition for output-observability of the parallel concatenated code obtained from these codes, as we can see in the following example.

Example 3.5.5. Let (A_1, B_1, C_1, D_1) with

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, D_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

and (A_2, B_2, C_2, D_2)

$$A_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}, D_2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$$

be the realizations of the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$.

Applying the test 3.2.1, obtained in chapter 2, it is easy to observe that these codes are not output-observable;

$$\text{rank } \begin{pmatrix} C_1 & D_1 \end{pmatrix} = \text{rank } \begin{pmatrix} 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 2 \end{pmatrix} = 1 < 2,$$
$$\text{rank } \begin{pmatrix} C_2 & D_2 \end{pmatrix} = \text{rank } \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{pmatrix} = 1 < 2.$$

Nevertheless, the parallel concatenated system is output observable, for that it suffices to observe that

$$\text{rank } (D_1 + D_2) = 2.$$

Neither a sufficient condition as we show in the following example.

Example 3.5.6. Let (A_1, B_1, C_1, D_1) with $A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $D_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and (A_2, B_2, C_2, D_2) with $A_2 = A_1$, $B_2 = B_1$, $C_2 = C_1$, $D_2 = -D_1$ be the realizations of the convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ respectively.

Clearly, both codes are output-observable:

$$\text{rank } \begin{pmatrix} C_1 & D_1 & & \\ C_1A_1 & C_1B_1 & D_1 & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 \end{pmatrix} = \text{rank } \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = 6$$

and

$$\text{rank } \begin{pmatrix} C_2 & D_2 & & \\ C_2A_2 & C_2B_2 & D_2 & \\ C_2A_2^2 & C_2A_2B_2 & C_2B_2 & D_2 \end{pmatrix} = \text{rank } \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} = 6.$$

However, the parallel concatenated code $\mathcal{C}(A, B, C, D,)$ where

$$A = \begin{pmatrix} 0 & 1 & & \\ 0 & 0 & & \\ & & 0 & 1 \\ & & 0 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \; D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

is not output observable, as we can see applying the test 3.2.1:

$$\operatorname{rank}\begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = 1 < 2.$$

Taking into account the above, we will try to find conditions for particular cases.

As a first result following corollary 3.2.2 we have:

Corollary 3.5.2. *Let $\mathcal{C}(A, B, C, D)$ be the parallel concatenated code obtained from the concatenation of the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$. If matrix $D_1 + D_2$ has full row rank, then the code $\mathcal{C}(A, B, C, D)$ is output observable.*

Working now, over a field $\mathbb{F}$ of characteristic different from 2, we consider a parallel concatenated code $\mathcal{C}(A, B, C, D)$ obtained from the concatenation of the codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, with $\mathcal{C}_2(A_2, B_2, C_2, D_2) = \mathcal{C}_1(A_1, B_1, C_1, D_1)$.

The output-observability matrix of this concatenated code is

$$\begin{pmatrix} C_1 & C_1 & 2D_1 & & & \\ C_1A_1 & C_1A_1 & 2C_1B_1 & 2D_1 & & \\ C_1A_1^2 & C_1A_1^2 & 2C_1A_1B_1 & 2C_1B_1 & 2D_1 & \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ C_1A_1^{2\delta_1} & C_1A_1^{2\delta_1} & 2C_1A_1^{2\delta_1-1}B_1 & \dots & 2C_1B_1 & 2D_1 \end{pmatrix}$$

and the rank of this matrix coincides with the rank of

$$\begin{pmatrix} C_1 & D_1 & & & \\ C_1A_1 & C_1B_1 & D_1 & & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 & \\ \vdots & \ddots & \ddots & & \\ C_1A_1^{2\delta_1} & C_1A_1^{2\delta_1-1}B_1 & \dots & C_1A_1B_1 & C_1B_1 & D_1 \end{pmatrix}$$

Notice that the submatrix

$$T_\delta = \begin{pmatrix} C_1 & D_1 & & & \\ C_1A_1 & C_1B_1 & D_1 & & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 & \\ \vdots & \ddots & \ddots & & \\ C_1A_1^{\delta_1} & C_1A_1^{\delta_1-1}B_1 & \dots & C_1A_1B_1 & C_1B_1 & D_1 \end{pmatrix}$$

corresponds to the output-observability matrix of the $\mathcal{C}_1(A, B, C, D)$ code.

Therefore, is having the following proposition.

Proposition 3.5.3. *Over a field $\mathbb{F}$ of characteristic different from two, a necessary condition for output-observability of the parallel concatenated code $\mathcal{C}(A, B, C, D)$ of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ with itself is that the code $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ be output-observable.*

Notice that over a field $\mathbb{F}$ of characteristic two the result is false.

Example 3.5.7. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ be a code with $A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $C_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $D_1 = \begin{pmatrix} 1 \end{pmatrix}$. This code is output observable but the parallel concatenated code $\mathcal{C}(A, B, C, D)$ of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ with itself is not output observable ($D = 2D_1 = 0$ and $CAB = 0$).

Calling T_i the matrix

$$T_i = \begin{pmatrix} C_1 & D_1 & & & \\ C_1A_1 & C_1B_1 & D_1 & & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & D_1 & \\ \vdots & \ddots & \ddots & & \\ C_1A_1^{i-1} & C_1A_1^{i-2}B_1 & \dots & C_1A_1B_1 & C_1B_1 \; D_1 \end{pmatrix}$$

for all $i \geq \delta$, we have the following theorem.

Theorem 3.5.1. *Suppose that the code $\mathcal{C}_1(A, B, C, D)$ is output observable. A necessary condition for output-observability of the concatenated code $\mathcal{C}(A, B, C, D)$ is*

$$\operatorname{rank} T_{\delta+1} - \operatorname{rank} T_\delta = n - k.$$

Proof. Following [25], for all $i \geq \delta$ the relation

$$\operatorname{rank} T_{i+1} - \operatorname{rank} T_i = \ell \,(\text{constant}).$$

□

Proposition 3.5.4. *Let $\mathcal{C}(A_1, B_1, C_1, D_1)$ and $\mathcal{C}(A_2, B_2, C_2, D_2)$ be two convolutional codes, such that $D_1 = D_2 = 0$, and $C_2 = 0$ (respec. $C_1 = 0$). Then, the parallel concatenated code $\mathcal{C}(A, B, C, D)$ is output observable if and only if $\mathcal{C}(A_1, B_1, C_1, D_1)$ (respc. $\mathcal{C}(A_2, B_2, C_2, D_2)$) is output observable.*

Proof. Taking into account $C = \begin{pmatrix} C_1 & 0 \end{pmatrix}$ and $D = 0$, we have that for all ℓ

$$\operatorname{rank} \begin{pmatrix} C_1 & 0 & 0 & & & & \\ C_1A_1 & 0 & C_1B_1 & 0 & & & \\ C_1A_1^2 & 0 & C_1A_1B_1 & C_1B_1 & 0 & & \\ \vdots & & & & \ddots & \ddots & \\ C_1A_1^\ell & 0 & C_1A_1^{\ell-1} & C_1A_1^{\ell-2}B_1 & \dots & C_1B_1 & 0 \end{pmatrix}$$

$$= \operatorname{rank} \begin{pmatrix} C_1 & 0 & 0 & & & \\ C_1A_1 & C_1B_1 & 0 & & & \\ C_1A_1^2 & C_1A_1B_1 & C_1B_1 & 0 & & \\ \vdots & & & \ddots & \ddots & \\ C_1A_1^\ell & C_1A_1^{\ell-1} & C_1A_1^{\ell-2}B_1 & \dots & C_1B_1 & 0 \end{pmatrix}.$$

□

Over a field $\mathbb{F}$ of characteristic different from two we have the following proposition.

Proposition 3.5.5. *Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be two convolutional codes with $(A_2, B_2, C_2, D_2) = (P^{-1}A_1P, P^{-1}B_1, C_1P, D_1)$ for some invertible matrix $P \in Gl(\delta, \mathbb{F})$. Then, the parallel concatenated code $\mathcal{C}(A, B, C, D)$ of these codes is output observable, if and only if the concatenated code $\mathcal{C}(\overline{A}, \overline{B}, \overline{C}, \overline{D})$ of $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ with itself is output observable.*

Proof. It suffices to observe that

$$\operatorname{rank} \begin{pmatrix} A_1 & 0 & B_1 & -I_1 & 0 & & & & & & & & \\ 0 & A_2 & B_2 & 0 & -I_1 & & & & & & & & \\ C_1 & C_2 & D_1+D_2 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & A_2 & B_2 & 0 & -I_1 & & & & & \\ 0 & 0 & 0 & C_1 & C_2 & D_1+D_2 & 0 & 0 & & & & & \\ & & & & & & \ddots & \ddots & \ddots & & & & \\ & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 & \\ & & & 0 & 0 & 0 & 0 & A_2 & B_2 & 0 & -I_1 & 0 & \\ & & & 0 & 0 & 0 & C_1 & C_2 & D_1+D_2 & 0 & 0 & 0 & \\ & & & & & & 0 & 0 & 0 & C_1 & C_2 & D_1+D_2 \end{pmatrix}$$

$$= \operatorname{rank} \mathbf{Q} \cdot \begin{pmatrix}
A_1 & 0 & B_1 & -I_1 & 0 & & & & & & \\
0 & A_1 & B_1 & 0 & -I_1 & & & & & & \\
C_1 & C_1 & 2D_1 & 0 & 0 & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & \\
0 & 0 & 0 & 0 & A_1 & B_1 & 0 & -I_1 & & & \\
0 & 0 & 0 & C_1 & C_1 & 2D_1 & 0 & 0 & & & \\
 & & & & & & \ddots & \ddots & \ddots & & \\
 & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & A_1 & B_1 & 0 & -I_1 & 0 \\
 & & & 0 & 0 & 0 & C_1 & C_1 & 2D_1 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & C_1 & C_1 & 2D_1
\end{pmatrix} \cdot \mathbf{P}$$

$$= \operatorname{rank} \begin{pmatrix}
A_1 & 0 & B_1 & -I_1 & 0 & & & & & & \\
0 & A_1 & B_1 & 0 & -I_1 & & & & & & \\
C_1 & C_1 & 2D_1 & 0 & 0 & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & & & \\
0 & 0 & 0 & 0 & A_1 & B_1 & 0 & -I_1 & & & \\
0 & 0 & 0 & C_1 & C_1 & 2D_1 & 0 & 0 & & & \\
 & & & & & & \ddots & \ddots & \ddots & & \\
 & & & 0 & 0 & 0 & A_1 & 0 & B_1 & -I_1 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & A_1 & B_1 & 0 & -I_1 & 0 \\
 & & & 0 & 0 & 0 & C_1 & C_1 & 2D_1 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & C_1 & C_1 & 2D_1
\end{pmatrix}$$

Where the matrices $\mathbf{Q}$ and $\mathbf{P}$ are:

$$\mathbf{Q} = \begin{pmatrix}
I_1 & & & & & & & & & & \\
 & P^{-1} & & & & & & & & & \\
 & & I_2 & & & & & & & & \\
 & & & I_1 & & & & & & & \\
 & & & & P^{-1} & & & & & & \\
 & & & & & I_1 & & & & & \\
 & & & & & & \ddots & & & & \\
 & & & & & & & P^{-1} & & & \\
 & & & & & & & & I_2 & & \\
 & & & & & & & & & I_1
\end{pmatrix}$$

and

$$\mathbf{P} = \begin{pmatrix} I_1 & & & & & & \\ & P & & & & & \\ & & I_2 & & & & \\ & & & I_1 & & & \\ & & & & \ddots & & \\ & & & & & P & \\ & & & & & & I_2 \end{pmatrix}$$

with I_1 and I_2 are the identity matrices in the adequate size. □

Example 3.5.8. Over $\mathbb{F}_5$, let (A, B, C, D) be a representation of a convolutional code with $A = (2)$, $B = (1)$, $C_1 = (3)$, $D_1 = (0)$.

The parallel concatenated code $\mathcal{C}(\overline{A}, \overline{B}, \overline{C}, \overline{D})$ is output-observable because matrices

$$T_\ell = \begin{pmatrix} 3 & 3 & 0 & & \\ 1 & 1 & 1 & 0 & \\ \vdots & & \ddots & & \\ 3 \cdot 2^\ell & 3 \cdot 2^\ell & & 1 & 0 \end{pmatrix}$$

has full row rank for all ℓ. Then, for all $P \in G(1, \mathbb{F})$ the parallel concatenated code of (A_1, B_1, C_1, D_1) with $(P^{-1}A_1P, P^{-1}B_1, C_1P, D_1)$ is output-observable. It is not difficult to observe that the output-observability matrix for this code is

$$\begin{pmatrix} 3 & 3p & 0 & & \\ 1 & 1p & 1 & 0 & \\ \vdots & & \ddots & & \\ 3 \cdot 2^\ell & 3 \cdot 2^\ell p & & 1 & 0 \end{pmatrix}$$

that has full row rank.

For fields with characteristic two the result fails. In fact we can deduce that

Proposition 3.5.6. *Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ be two convolutional codes with $(A_2, B_2, C_2, D_2) = (P^{-1}A_1P, P^{-1}B_1, C_1P, D_1)$ for some invertible matrix $P \in Gl(\delta, \mathbb{F})$. Then, the parallel concatenated code $\mathcal{C}(A, B, C, D)$ of these codes is never output observable.*

Proof. The output-observability matrix for this case is

$$T_\ell = \begin{pmatrix} C_1 & C_1P & 0 & & \\ C_1A_1 & C_1A_1P & 0 & 0 & \\ \vdots & \vdots & & \ddots & \\ C_1A_1^\ell & C_1A_1^\ell P & 0 & 0 & \dots \quad 0 \end{pmatrix}.$$

□

3.5.3 Systematic serial concatenation

In order to analyze the output-observability of the systematic serial concatenated code, we compute the rank of the matrices M_ℓ

$$M_\ell = \begin{pmatrix} A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & & & & \\ B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & & & & & & \\ C_1 & 0 & D_1 & 0 & 0 & & & & & & \\ D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & \\ 0 & 0 & 0 & B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & & & \\ 0 & 0 & 0 & C_1 & 0 & D_1 & 0 & 0 & & & \\ 0 & 0 & 0 & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & \\ & & & & & \ddots & & & & & \\ & & & & & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & \\ & & & & & B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & \\ & & & & & C_1 & 0 & D_1 & 0 & 0 & \\ & & & & & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & \\ & & & & & & & & C_1 & 0 & D_1 \\ & & & & & & & & D_2C_1 & C_2 & D_2D_1 \end{pmatrix}.$$

Example 3.5.9. Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ and $C_2(A_2, B_2, C_2, D_1)$ the realizations of two convolutional codes with

$$A_1 = (0),\ B_1 = \begin{pmatrix} 1 & 0 \end{pmatrix},\ C_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

$$A_2 = (1),\ B_2 = \begin{pmatrix} 0 & 1 \end{pmatrix},\ C_2 = (1), D_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}.$$

Then the systematic serial concatenated code is $\mathcal{C}(A, B, C, D)$ with

$$A = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix},\ B = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix},\ D = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

The system is not output-observable because of

$$\begin{aligned}
\operatorname{rank} M_0 &= \operatorname{rank} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} = 3 \\
\operatorname{rank} M_1 &= \operatorname{rank} \begin{pmatrix} 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = 6 < 8
\end{aligned}$$

Observe that the first system is not output-observable:

$$\begin{aligned}
\operatorname{rank} M_0 &= \operatorname{rank} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} = 2 \\
\operatorname{rank} M_1 &= \operatorname{rank} \begin{pmatrix} 1 & 0 & 0 & & \\ 1 & 0 & 1 & & \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = 3 < 4.
\end{aligned}$$

Proposition 3.5.7. *A necessary condition for output-observability of systematic serial concatenated system is that the code $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ be output-observable.*

Proof. Making block row transformations to the matrix M_ℓ we have that

$$\operatorname{rank} M_\ell = \operatorname{rank} \begin{pmatrix}
A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & & & & \\
C_1 & 0 & D_1 & 0 & 0 & & & & & & \\
0 & 0 & 0 & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & \\
0 & 0 & 0 & C_1 & 0 & D_1 & 0 & 0 & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & \\
 & & & & & C_1 & 0 & D_1 & 0 & 0 & \\
 & & & & & & & & C_1 & 0 & D_1 \\
B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & & & & & & \\
D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & & & & \\
0 & 0 & 0 & B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & & & \\
0 & 0 & 0 & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & B_2C_1 & A_2 & B_2D_1 & 0 & -I_{\delta_2} & \\
 & & & & & D_2C_1 & C_2 & D_2D_1 & 0 & 0 & \\
 & & & & & & & & D_2C_1 & C_2 & D_2D_1
\end{pmatrix}.$$

□

3.5.4 Parallel interleaver concatenation

In order to obtain conditions for the output-observability character of a parallel interleaver concatenated code, we consider the output-observability matrix M_ℓ.

Proposition 3.5.8. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, and $\mathcal{P}$ an interleaver matrix. A sufficient condition for output-observability of the concatenated interleaver code of both codes is that the matrix $D_1 + D_2\mathcal{P}$ have full rank.*

Proposition 3.5.9. *Let (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) be realizations of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, and $\mathcal{P}$ the interleaver matrix. Then the parallel interleaver concatenated system of both codes is output observable if and only if the parallel concatenated code of $(A_1, B_1\mathcal{P}^{-1}, C_1, D_1\mathcal{P}^{-1})$ and (A_2, B_2, C_2, D_2) is.*

Proof. If we compute the output-observability character of the parallel inter-

leaver concatenated code, we can observe that:

$$\text{rank}\begin{pmatrix} A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & & \dots & & 0 & 0 \\ 0 & A_2 & B_2\mathcal{P} & 0 & -I_{\delta_2} & & & & & & & \\ C_1 & C_2 & D_1+D_2\mathcal{P} & 0 & 0 & & & & & & & \\ & & & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & & & & \\ & & & 0 & A_2 & B_2\mathcal{P} & 0 & -I_{\delta_2} & & & & \\ & & & C_1 & C_2 & D_1+D_2\mathcal{P} & 0 & 0 & & & & \\ \vdots & & & & & & \ddots & & & & & \\ & & & & & \dots & A_1 & 0 & B_1 & -I_{\delta_1} & 0 & 0 \\ & & & & & \dots & 0 & A_2 & B_2\mathcal{P} & 0 & -I_{\delta_2} & 0 \\ & & & & & \dots & 0 & 0 & 0 & C_1 & C_2 & D_1+D_2\mathcal{P} \end{pmatrix}$$

$=$

$$\text{rank}\begin{pmatrix} A_1 & 0 & B_1\mathcal{P}^{-1} & -I_{\delta_1} & 0 & & & & \dots & & 0 & 0 \\ 0 & A_2 & B_2 & 0 & -I_{\delta_2} & & & & & & & \\ C_1 & C_2 & D_1\mathcal{P}^{-1}+D_2 & 0 & 0 & & & & & & & \\ & & & A_1 & 0 & B_1\mathcal{P}^{-1} & -I_{\delta_1} & 0 & & & & \\ & & & 0 & A_2 & B_2 & 0 & -I_{\delta_2} & & & & \\ & & & C_1 & C_2 & D_1\mathcal{P}^{-1}+D_2 & 0 & 0 & & & & \\ \vdots & & & & & & \ddots & & & & & \\ & & & & & \dots & A_1 & 0 & B_1\mathcal{P}^{-1} & -I_{\delta_1} & 0 & 0 \\ & & & & & \dots & 0 & A_2 & B_2 & 0 & -I_{\delta_2} & 0 \\ & & & & & \dots & 0 & 0 & 0 & C_1 & C_2 & D_1\mathcal{P}^{-1}+D_2 \end{pmatrix}.\mathbf{P}$$

with

$$\mathbf{P} = \begin{pmatrix} I_{\delta_1} & & & & & \\ & I_{\delta_2} & & & & \\ & & \mathcal{P} & & & \\ & & & \ddots & & \\ & & & & I_{\delta_2} & \\ & & & & & \mathcal{P} \end{pmatrix}.$$

□

Chapter 4

Decoding problem

In this chapter, we briefly recall notions already known on the decoding, and in particular algorithms for decoding already known such as the Viterbi, or the Berlekamp-Massey for instance. We also implement our own decoding algorithms inspired by the linear systems theory approach for convolutional codes as presented in previous chapters. Our approach is an iterative method, which suggests decoding step by step for each state of the process. Apart from the general method, we present as well derived methods specially suitable to each model of concatenation that is involved with our work. Material of this chapter can be found in [29].

4.1 Introduction

It is well known the existence of several algorithms for decoding convolutional codes. Foremost among them, codes are decoded using the so called Viterbi decoding algorithm. The Viterbi Algorithm was first proposed as a solution to the decoding of convolutional codes by Andrew J. Viterbi in 1967 [84]. With regard to the case of concatenated codes, some decoding methods have also been suggested as in [7]. When it comes to the linear systems approach for the decoding algorithm, an algebraic method has also been suggested by J. Rosenthal in [67], which calls upon the controllability and observability properties of convolutional codes, as well as algorithms such as the Berlekamp-Massey one [54]. We try to implement algorithms better suited to our own linear systems construction.

In order to analyze this process we assume that a certain code word $\{v_t\}_{t\geq 0} = \left\{\begin{pmatrix} u_t \\ y_t \end{pmatrix}\right\}$ is sent and the message word $\{\widehat{v}_t\}_{t\geq 0} = \left\{\begin{pmatrix} \widehat{u}_t \\ \widehat{y}_t \end{pmatrix}\right\}$ is received. The decoding problem then asks for the minimization of the error

$$\begin{array}{rl} \text{error} & = \min_{\{v_t\}\in C} \sum_{t=0}^{\infty} \text{dist}\,(v_t, \widehat{v}_t) \\ & = \min\,(\sum_{t=0}^{\infty}(\text{dist}\,(y_t, \widehat{y}_t) + \text{dist}\,(u_t, \widehat{u}_t)) \end{array} \tag{4.1}$$

where the weight

$$w(\{v_t\}_{t>0}) = \sum_{t=0}^{\infty}(w(y_t) + w(u_t)$$

and

$$\text{dist}\,(v_t, \widehat{v}_t) = w(v_t - \widehat{v}_t).$$

If in the transmission, no errors are produced, then $\{\widehat{v}_t\}_{t\geq 0}$ is a valid trajectory and the error value defined in (4.1) is zero.

Otherwise, if the error value is not null, then the sequence received is not a codeword, and does not belong to the code family. Then, comes the importance of decoding, which consists of finding out, from the gotten sequence the encoded word supposed to have been received.

However, it is also possible to consider the transmission done over the "Gaussian channel"; indeed, it is possible to give conditions for minimization of the error in the case when we do the decoding on $\mathbf{R}$ linear trajectories [67]. In this case, let us consider a convolutional code $\mathcal{C}$ described by the realization (A, B, C, D) and let $T > \Theta$ integers satisfying the following assumptions.

Assumptions 4.1.1. *i) A is invertible, the matrix $\begin{pmatrix} B & AB & \dots & A^{T-1}B \end{pmatrix}$ has full row rank δ and its rows form the parity check matrix of a block code of distance at least d_1.*

ii) The matrix $\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\Theta-1} \end{pmatrix}$ has full column rank δ and its rows form the generator matrix of a block code of distance d_2.

Remark 10. These conditions imply that the pair (A, B) is controllable and T is necessarily larger than controllability index, and the pair (A, C) is observable and Θ is necessarily larger than observability index.

Under these conditions, we have the following lemma:

Lemma 4.1.1. *[67] Assume that the matrices A, B, C and the integers T, Θ satisfy the assumptions 4.1.1. Assume that*

$$\begin{pmatrix} u(t) \\ y(t) \end{pmatrix}_{t\geq 0} \text{ and } \begin{pmatrix} \tilde{u}(t) \\ \tilde{y}(t) \end{pmatrix}_{t\geq 0}$$

are two sets of codewords both satisfying the conditions above. Let $x(t)_{t\geq 0}$ and $\tilde{x}(t)_{t\geq 0}$ be the corresponding set of state vectors. If there is a $\tau \geq 0$ with

$$x_\tau = \tilde{x}_\tau \text{ and } x_{\tau+1} \neq \tilde{x}_{\tau+1}$$

then for any γ satisfying $\tau + T > \gamma \geq \tau$ one has that

$$\sum_{t=\tau}^{\gamma} (\operatorname{dist}(u(t), \tilde{u}(t)) + \operatorname{dist}(y(t), \tilde{y}(t))) \geq \min(d_1, \lfloor \frac{\gamma - \tau}{\Theta} \rfloor + 1)$$

4.2 Decoding convolutional codes

We are interested in the decoding of convolutional codes represented as linear systems.

In general, using the matrix (5.4) we obtain a representation in terms of state input-output of the code

$$\begin{pmatrix} C & D & & & & \\ CA & CB & D & & & \\ CA^2 & CAB & CB & D & & \\ \vdots & & & \ddots & \ddots & \\ CA^\ell & CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \\ \vdots \\ u(\ell) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(\ell) \end{pmatrix} \tag{4.2}$$

Proposition 4.2.1. *Let (A, B, C, D) be a representation of an output-observable code. Then, the system* (4.2) *is solvable.*

Proof. Assume the system (A, B, C, D) is output-observable; then the matrix of the equation (4.2) has full row rank. □

Remark 11. It is usual to consider the initial state of the system $x(0) = 0$. In this case the system (4.2) is reduced to

$$\begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ \vdots \\ u(\ell) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(\ell) \end{pmatrix}. \tag{4.3}$$

So, in this case the solvability of the system is ensured if the matrix

$$\widehat{T}_{\ell-1} = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}$$

has full rank.

But, if the matrix of the system (4.2) has full row rank, the system (4.3) is not necessarily solvable as we can see in the following example.

Example 4.2.1. Let (A, B, C, D) a realization of a convolutional code with $A = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $C = \begin{pmatrix} 0 & 1 \end{pmatrix}$ and $D = (0)$, the system

$$\begin{pmatrix} C & D & \\ CA & CB & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$$

is compatible for all $\begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$ and the solution is $x_1 = y(1)$, $x_2 = y(0)$, nevertheless the system

$$\begin{pmatrix} D & \\ CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$$

has only solution for $y(0) = y(1) = 0$. That is to say the initial condition for the system are restrictive conditions for solving the system.

But, in any case, we have the following proposition.

Proposition 4.2.2. *If the matrix $\widehat{T}_{\ell-1}$ has full row rank the system* (4.2) *is solvable with initial condition $x(0) = 0$.*

Proof. If the matrix $\widehat{T}_{\ell-1}$ has full row rank, the system (4.3) is solvable. Then, if $(u(0), \ldots, u(\ell))$ is a solution of this system, clearly $(0, u(0), \ldots, u(\ell))$ is a solution for the system (4.2). □

Example 4.2.2. In $\mathbb{F}_2$, let (A, B, C, D) be a representation of the convolutional code $\mathcal{C}$ with

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ B = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 \end{pmatrix},\ D = \begin{pmatrix} 1 \end{pmatrix}$$

Clearly $\det(zI - A) = z^2 + 1$.

Let be now $m = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \end{pmatrix}$, so $m(z) = 1 + z + z^3 + z^4$,

and $(1 + z^2)m(z) = (1 + z + z^2 + z^4 + z^5 + z^6)$;

then the inputs are

$$(u(0), u(1), u(2), u(3), u(4), u(5), u(6)) = (1, 1, 1, 0, 1, 1, 1)$$

written in a polynomial form $1 + 1s + 1s^2 + 0s^3 + 1s^4 + 1s^5 + 1s^6$

$$G(z)m(z) = \begin{pmatrix} P(z)Q^{-1} \\ I \end{pmatrix} Q(z)m(z) = \begin{pmatrix} C(zI - A)^{-1}B + D \\ I \end{pmatrix} Q(z)m(z)$$

$$= \begin{pmatrix} 1 + z^6 \\ 1 + z + z^2 + z^4 + z^5 + z^6 \end{pmatrix},$$

$Q(s)m(z) = 1 + z + z^2 + z^4 + z^5 + z^6)$.

Suppose that the initial state is $x(0) = 0$, the outputs $(y(0), y(1),\ y(2),\ y(3),\ y(4),\ y(5), y(6))$ can be obtained in the following manner:

$$y(0) = Cx(0) + Du(0) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1 \cdot 1 = 1$$

$$x(1) = Ax(0) + Bu(0) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$y(1) = Cx(1) + Du(1) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 1 \cdot 1 = 1 + 1 = 0$$

$$x(2) = Ax(1) + Bu(1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$y(2) = Cx(2) + Du(2) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 \cdot 1 = 0$$
$$x(3) = Ax(2) + Bu(2) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$y(3) = Cx(3) + Du(3) \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 1 \cdot 0 = 0 + 0 = 0$$
$$x(4) = Ax(3) + Bu(3) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$y(4) = Cx(4) + Du(4) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 1 \cdot 1 = 0$$
$$x(5) = Ax(4) + Du(4) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$y(5) = Cx(5) + Du(5) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 \cdot 1 = 1 + 1 = 0$$
$$x(6) = Cx(5) + Du(5) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$y(6) = Cx(6) + Du(6) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 1 \cdot 1 = 0 + 1 = 1$$

and the outputs are $(1, 0, 0, 0, 0, 0, 1)$.

Reciprocally

Given $(y(0), y(1), y(2), y(3), y(4), y(5), y(6)) = (1, 0, 0, 0, 0, 0, 1)$, solving the system

$$\begin{pmatrix} C & D & & & & & & \\ CA & CB & D & & & & & \\ CA^2 & CAB & CB & D & & & & \\ CA^3 & CA^2B & CAB & CB & D & & & \\ CA^4 & CA^3B & CA^2B & CAB & CB & D & & \\ CA^5 & CA^4B & CA^3B & CA^2B & CAB & CB & D & \\ CA^6 & CA^5B & CA^4B & CA^3B & CA^2B & CAB & CB & D \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ u(0) \\ u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ u(6) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Concretely:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ u(0) \\ u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ u(6) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The solution is

$$\begin{aligned} x_1 &= 0u(5) + 1u(6) + 1 \\ x_2 &= 1u(5) + 0u(6) + 1 \\ u(0) &= 0u(5) + 1u(6) + 0 \\ u(1) &= 1u(5) + 1u(6) + 1 \\ u(2) &= 1u(5) + 0u(6) + 0 \\ u(3) &= 0u(5) + 1u(6) + 1 \\ u(4) &= 1u(5) + 1u(6) + 1 \end{aligned}$$

Taking into account the initial states are $x_1 = 0$, $x_2 = 0$, we have

$$\begin{aligned} (u(0), u(1), u(2), u(3), u(4), u(5), u(6)) &= (1, 1, 1, 0, 1, 1, 1) \\ &= 1 + z + z^2 + z^4 + z^5 + z^6 \\ &= Q(z)m(z) \\ &= (1 + z^2)m(z) \end{aligned}$$

then $m(z) = 1 + z + z^3 + z^4 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Describing the code vector $v(t)$

System (4.3) permits us to describe the code vector $v(t) = \begin{pmatrix} u(t) \\ y(t) \end{pmatrix}$ as a solution of a linear homogeneous equation, as we have seen in Theorem 3.2.1.

Proposition 4.2.3. *The code vectors $v(t)$ of a convolutional code (A, B, C, D) with initial state $x(0) = 0$ are the solutions of the following linear homogeneous*

system

$$\begin{pmatrix} D & -I & 0 & 0 & & & & & & \\ CB & 0 & D & -I & & & & & & \\ CAB & 0 & CB & 0 & D & -I & & & & \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \ddots & & & \\ CA^{\ell-1}B & 0 & CA^{\ell-2}B & & & & CB & 0 & D & -I \end{pmatrix} \begin{pmatrix} u(0) \\ y(0) \\ u(1) \\ y(1) \\ \vdots \\ u(\ell) \\ y(\ell) \end{pmatrix} = 0 \tag{4.4}$$

Proof. For that, we observe that (4.3) can be written in the following manner

$$\begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ \vdots \\ u(\ell) \end{pmatrix} = \begin{pmatrix} I & & \\ & I & \\ & & \ddots \\ & & I \end{pmatrix} \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(\ell) \end{pmatrix}$$

equivalently

$$\begin{pmatrix} D & 0 & 0 & 0 & -I & & & \\ CB & D & & & & -I & & \\ CAB & CB & D & & & & -I & \\ \vdots & \vdots & & \ddots & \ddots & & & \\ CA^{\ell-1}B & CA^{\ell-2}B & & CB & D & & & -I \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ \vdots \\ u(\ell) \\ y(0) \\ y(1) \\ \vdots \\ y(\ell) \end{pmatrix} = 0,$$

now, we just need to reorder the columns of the matrix as well as the rows of the vector and then the proof is completed. □

Example 4.2.3. In $\mathbb{F}_5$, we consider $A = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 \end{pmatrix}$, $D = \begin{pmatrix} 3 \end{pmatrix}$.

We consider the sequence $v = (v(0), v(1), v(2))$ with $v(0) = (1,\ 3)$, $v(1) = (2,\ 4)$, $v(2) = (3,\ 1)$

We have $\ell = 2$, which means that our matrix is:

$$\begin{pmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 3 & -1 & 0 & 0 \\ 2 & 0 & 4 & 0 & 3 & -1 \end{pmatrix}$$

For sequences of $\ell = 2$, the set of solutions of the homogeneous equation is such that:

$$\begin{pmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 3 & -1 & 0 & 0 \\ 2 & 0 & 4 & 0 & 3 & -1 \end{pmatrix} \begin{pmatrix} u(0) \\ y(0) \\ u(1) \\ y(1) \\ u(2) \\ y(2) \end{pmatrix} = 0.$$

Then, the possible code vectors $v(t)$ are the elements of vector subspace

$$\left\{ \begin{pmatrix} u(0) \\ 3u(0) \end{pmatrix}, \begin{pmatrix} u(1) \\ 4u(0) + 3u(1) \end{pmatrix}, \begin{pmatrix} u(2) \\ 2u(0) + 4u(1) + 3u(2) \end{pmatrix}, \forall u(0), u(1), u(2) \in \mathbb{F}_5 \right\}$$

Now, we verify whether our vector code v is a solution to the previous system.

We easily check that:

$$\begin{pmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 3 & -1 & 0 & 0 \\ 2 & 0 & 4 & 0 & 3 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \\ 4 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} \neq 0$$

Finally, v is not a solution to the homogeneous system, then v is not a valid vector code.

A valid vector code with outputs $y(0) = 3$, $y(1) = 4$, $y(2) = 1$ can be obtained solving the system

$$\begin{cases} 3 &= 3u(0) \\ 4 &= 4u(0) + 3u(1) \\ 1 &= 2u(0) + 4u(1) + 3u(2) \end{cases}$$

the solution is $u(0) = 1$, $u(1) = 0$, $u(2) = 3$.

Then, the vector code with outputs $y(0) = 3$, $y(1) = 4$, $y(2) = 1$ is $\overline{v} = ((1,3),(0,4),(3,1))$ and $d(v,\overline{v}) = 1$.

Analogously, a valid vector code with entries $u(0) = 1$, $u(1) = 2$, $u(2) = 3$ can be obtained directly as follows

$$\begin{cases} y(0) &= 3u(0) = 3 \\ y(1) &= 4u(0) + 3u(1) = 0 \\ y(2) &= 2u(0) + 4u(1) + 3u(2) = 4 \end{cases}$$

Then, the valid vector code with entries $u(0) = 1$, $u(1) = 2$, $u(2) = 3$ is $\overline{\overline{v}} = ((1,3),(2,0),(3,4))$ and $d(v,\overline{\overline{v}}) = 2$.

Now, we compute the free distance.

Since the free distance is equivalent to the minimum weight of all codewords (except for the zero word), it can also be found in the case when almost all inputs are null.

In other words, in our case, taking into account the triangular shape having the solution, if we consider for $i \in \{0,\ldots,\ell-1\}$ that $u(i) = 0$, then $y(i) = 0$, $0 \leq i \leq \ell - 1$; and if we consider $u(\ell) \neq 0$ we have that $v(\ell)$ has minimum weight. In our particular case, if we have $u = (u(0),u(1),u(2)) = (0,0,x)$ with $x \in \mathbb{F}_5,\ x \neq 0$. Then,

$$\begin{pmatrix} v(0) \\ v(1) \\ v(2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ x \\ 3x \end{pmatrix}$$

the code vector $(v(0),v(1),v(2))$ has minimum weight, which means that the free distance is 2.

Remark 12. If $u(0) = u(1) = \ldots = u(\ell-1) = 0$ then $y(0) = y(1) = \ldots = y(\ell-1) = 0$, then the weight w verifies $w \leq k+p$. And if we consider $u(\ell) = e_j$ the j-vector of the canonical basis we have that the weight w verifies $w \leq 1+p$.

Solving the system (4.2)

In the case where the matrix of the system (4.2) does not have full row rank, the existence of the solution is not guaranteed and depends on $\binom{y(0)}{y(1)}$.

If the system is not compatible we can find approximate solutions using generalized inverse matrices and under some conditions we can consider the Moore-Penrose pseudoinverse matrix.

Remember that, given a matrix $A \in M_{n\times m}(\mathbb{F})$, a matrix $X_A \in M_{m\times n}(\mathbb{F})$ is called generalized inverse if it satisfies

a) $AX_AA = A$,

A generalized inverse X_A of A is called a reflexive generalized inverse if it satisfies

b) $X_AAX_A = X_A$

A reflexive generalized inverse X_A of A is called normalized and will be denoted by A^{nor} if it satisfies

c) $(AA^{nor})^t = AA^{nor}$

And finally, a normalized generalized inverse A^{nor} is called the Moore-Penrose pseudoinverse and will be denoted by A^+ if it satisfies

d) $(A^+A)^t = A^+A$.

Obviously, if A is a square invertible matrix, then the matrix X_A exists and $X_A = A^{-1}$.

A linear system $Ax = y$ can be solved if we have a generalized inverse of the matrix A.

Observe that if

$$Ax = y$$

we have

$$AX_AAx = y$$

so

$$AX_Ay = y$$

that is to say

$$X_Ay$$

is a solution and the general solution can be easily obtained if we are taking into account that $\mathrm{Im}\,(I_m - X_A A) = \mathrm{Ker}\, A$, that is to say, if $X_A y$ is a solution, then $X_A y + (w - X_A A w)$ is also a solution.

If does not exists a solution x of the system, $X_A y$ is an approximated solution of the system (i.e. is a solution of the compatible system $Ax = AX_A y = \overline{y}$).

Not always there exists the normalized and pseudoinverse matrix. Penrose [14] showed that every matrix A over the complex field has a normalized inverse and a unique A^+. However, Pearl [13] showed that a matrix $A \in M_{m\times n}(\mathbb{F})$ of rank r over an arbitrary field has a normalized and a Moore-Penrose A^+ (unique) only under certain conditions. In fact we have the following result

Theorem 4.2.1. *Let A be an $m \times n$ matrix of rank r over a field F (having involutory automorphism). Then, A has a normalized generalized inverse A^{nor} if and only if*

$$r = \mathrm{rank}\,(A^t A).$$

And A has a Moore-Penrose pseudoinverse A^+ if and only if

$$r = \mathrm{rank}\,(A^t A) = \mathrm{rank}\,(A A^t).$$

Example 4.2.4. Over $\mathbb{F}_5$ the 1×5-matrix $A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ does not have a A^+ pseudo-inverse.

Anyway, we have the following result (see [91] for more details).

Lemma 4.2.1. *For any matrix over an arbitrary field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime and $m \geq 1$, there exists a reflexive generalized inverse matrix.*

For those methods of solving, instead of only detecting the error, (as for instance in the Viterbi decoding algorithm), and put out the correct sequence that should have been received, at the same time we detect the error, and give the original message before encoding.

Proposition 4.2.4. *Let $\mathbb{F}_q$ be a field with $q = q_1^m$, and consider a representation (A, B, C, D) of a convolutional code. Suppose that $y = (y(0), \ldots, y(\ell))$ is a received sequence, we have that $u = (u(0), u(1) \ldots, u(\ell))$ is obtained as follows*

a) Set $x(0) = 0$, then $u = X_{\widehat{T}_\ell} y$

b) Do not set $x(0)$, then $u = X_{T_\ell} y$.

Example 4.2.5. In $\mathbb{F}_2$, let us consider the code $\mathcal{C}(A, B, C, D)$ defined by:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \; C = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

We observe that the system is output-observable.

Suppose that we want to decode the sequence

$$y = (y(0), y(1), y(2)) = (1\,0,\ 0\,1,\ 0\,1)$$

We set $x(0) = (0\,0)$

Then, we solve

$$\begin{pmatrix} D & & \\ CB & D & \\ CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \\ y(2) \end{pmatrix}$$

In this case

$$X_{\widehat{T}_\ell} = \widehat{T}_\ell = \begin{pmatrix} 0 & 1 & & & & \\ 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & & \\ 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

and the decoded sequence is $u = (u(0), u(1), u(2)) = (0\,1\ 1\,0\ 1\,0)$.

4.3 The first iterative decoding algorithm

Our contribution can be found at [29]. In order to easily solve the system (4.2), the algorithms we will be using are divided between some steps.

The first algorithm for solving focuses more on directly correcting the error in case of disturbance by approaching the original input that is supposed to have been encoded (and whose encoding was disturbed, in case of disturbance). We consider that if we get as close as possible to the initial input, with the

output received, by approaching the best possible solution of our system, then we will have the error implicitly corrected and the original message. We do so considering initial conditions, and output-observability matrix.

Of course, initial conditions are not our main concern. We denote by X_M a generalized inverse of the matrix M, and by M^+ the Moore-Penrose pseudoinverse of the matrix M, if exists.

Step 1: Look at the number of outputs denoted by ℓ; give ℓ

Step 2: Solve

$$T_0 \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = \begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0) \tag{4.5}$$

If $\operatorname{rank}(C\ D)$ is row maximal (4.5) is compatible; in particular if $\operatorname{rank} D$ is row maximal, then (4.5) is compatible as well.

switch case 1: Fix $x(0)$ such that $x(0) = 0$ then solve: $Du(0) = y(0)$
If D^+exists $D^+\big(y(0)\big)$ is a solution; otherwise $X_D\big(y(0)\big)$ is

case 2: Do not fix $x(0)$ then solve $\begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$
If T_0^+exists $T_0^+\big(y(0)\big)$ is a solution; otherwise $X_{(C\ D)}\big(y(0)\big)$ is

Then, we find approximate solutions by computing the Moore-Penrose generalized inverse of $\widehat{T_0}$ considering $x(0) = 0$ or T_0 for no fixed value for $x(0)$ and minimize $d_H(T_0 \begin{pmatrix} x(0) \\ u(0) \end{pmatrix}, y(0))$ and Hamming weight of u as well. Then, settle for the approximate minimal solution.

Step 3: Iteratively, solve $T_\ell \begin{pmatrix} x(0) \\ u(0) \\ \vdots \\ u(\ell) \end{pmatrix} = \begin{pmatrix} y(0) \\ \vdots \\ y(\ell) \end{pmatrix}$, for all ℓ

Considering the list of solutions $x(0)$, $u(0)$, ..., $u(\ell - 1)$, we solve

$$Du(\ell) = y(\ell) - CA^\ell x(0) - CA^{\ell-1}Bu(0) - \ldots - CBu(\ell - 1)$$

and

$$u(\ell) = D^+(y(\ell) - CA^\ell x(0) - CA^{\ell-1}Bu(0) - \ldots - CBu(\ell - 1)$$

is a solution if D^+ exists; otherwise

$$X_D(y(\ell) - CA^\ell x(0) - CA^{\ell-1}Bu(0) - \ldots - CBu(\ell-1)$$

is a solution, for all ℓ. As before, minimize $d_H(T_\ell \begin{pmatrix} x(0) \\ u \end{pmatrix}, y)$ and Hamming weight of $u = (u(0), \ldots, u(\ell))$ as well. Settle for that approximate minimized solution.

Example 4.3.1. In $\mathbb{F}_2$, let us consider the code $\mathcal{C}(A, B, C, D)$ defined by

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \ C = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

First of all we observe that the system is output-observable because the matrix D has full row rank.

Suppose that we want to decode the sequence:

$$y = (y(0), y(1), y(2)) = (1\,0,\ 0\,1,\ 0\,1)$$

Step 1: We have $\ell = 2$

Step 2: We have

$$D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix};$$

then we solve (4.5) $T_0 \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$

We have that rank $D = 2$, D has full rank; thus, we decide to choose the usual initial state $x(0) = (0\ 0)$; so we solve

$\widehat{T}_0(u_0) = Du(0) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \big(u(0)\big) = y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

We easily get $u(0) = (0\,1)$

Step 3: We solve $T_1 \begin{pmatrix} x(0) \\ u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$ with $x(0) = (0\,0)$ and $u(0) = (0\ 1)$

Then, it suffices to solve

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \big(y(1)\big), \quad \text{with } u(0) = (0\,1)$$

that is to say

$$Du(1) = y(1) - CBu(0)$$

$$u(1) = D^{+}(y(1) - CBu(0)) = D^{+}y(1)$$

We obtain $u(1) = (1\,0)$

Finally, we solve $T_2 \begin{pmatrix} x(0) \\ u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2)$ with $x(0) = (0\,0)$, $u(0) = (0\,1)$ and $u(1) = (1\,0)$

Then, it suffices to solve

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2), \text{ with } u(0) = (0\,1), \text{ and } u(1) = (1\,0)$$

that is to say

$$Du(2) = y(2) - CABu(0) - CBu(1)$$

So, we easily obtain $u(2) = (1\,0)$

For this case, $y = (y(0), y(1), y(2)) = (1\,0\ 0\,1\ 0\,1)$ there was no error during transmission;

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (0\,1\ 1\,0\ 1\,0)$

Example 4.3.2. Over the field $\mathbb{F}_3$, let us consider the code $\mathcal{C}(A, B, C, D)$ given by :

$$A = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}, \ C = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}$$

First of all we observe that the system is not output-observable because of $\operatorname{rank} T_2 < 6$.

When we try to decode the sequence $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,1,\ 0\,2)$, the steps we follow are the following:

Step 1: We have $\ell = 2$

Step 2: We have

$$D = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix};$$

We have that $\operatorname{rank} D = 1 < 2 = p$, then, the existence of a solution of the system $Du(0) = y(0)$ depends on $y(0)$, in our particular case the system is not solvable, however, $\operatorname{rank}\begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank}\begin{pmatrix} 1 & 1 & 0 & 2 \\ 2 & 0 & 0 & 0 \end{pmatrix} = 2 = p$, so, we choose not to fix $x(0)$.

So, we solve $T_0 \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$

$$\begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 2 \\ 2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} = y(0)$$

Taking into account that

$$\operatorname{rank}\begin{pmatrix} C & D \end{pmatrix} = \operatorname{rank}\begin{pmatrix} C & D \end{pmatrix}\begin{pmatrix} C & D \end{pmatrix}^t = \operatorname{rank}\begin{pmatrix} C & D \end{pmatrix}^t\begin{pmatrix} C & D \end{pmatrix} = 2$$

there exists the generalized inverse of such a matrix $M = \begin{pmatrix} C & D \end{pmatrix}$ that is given by $M^+ = M^t(MM^t)^{-1}$. So, we get: $(C\ D)^+ = \begin{pmatrix} 0 & 2 \\ 2 & 2 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$ Then,

$$\begin{pmatrix} 0 & 2 \\ 2 & 2 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Indeed, we get: $x(0) = (2\ 0)$ and $u(0) = (0\ 0)$.

Notice that $x(0) = (2\,0)$ and $u(0) = (0\,0)$ is a particular solution of the compatible system $\begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$. In fact, all solutions are $x(0) = (2\ 0) + \alpha(0\ 2)$ and $u(0) = (0\ 0) + \alpha(0\ 2) + \beta(2\ 0)$.

Step 3: Solve $T_1 \begin{pmatrix} x(0) \\ u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$ with $x(0) = (2\ 0)$ and $u(0) = (0\ 0)$.

It suffices to solve $\begin{pmatrix} CA & CB & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \\ u(1) \end{pmatrix} = y(1)$ with $x(0) = (2\ 0)$ and $u(0) = (0\ 0)$. Equivalently, we solve

$$Du(1) = y(1) - CAx(0) - CBu(0).$$

We get $Du(1) = y(1) - \begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix}$;

Taking into account that

$$\operatorname{rank} D = \operatorname{rank} DD^t = \operatorname{rank} D^tD = 1,$$

there exists $D^+ = \begin{pmatrix}0 & 0\\0 & 2\end{pmatrix}$. So,

$D^+ \begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0 & 0\\2 & 0\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0\\2\end{pmatrix}$ is an approximate solution for $u(1)$.

Since $\begin{pmatrix}1 & 2 & 2 & 1 & 0 & 2\\0 & 1 & 0 & 2 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\\0\\2\end{pmatrix} - y(1) = \begin{pmatrix}0\\2\end{pmatrix} \neq \begin{pmatrix}0\\0\end{pmatrix}$; we can detect that there was an error produced at the second element of the sequence $y(1) = (0\ 1)$, and our approximate solution is $u(1) = (0\,2)$.

Finally, we solve $T_2\begin{pmatrix}x(0)\\u(0)\\u(1)\\u(2)\end{pmatrix} = \begin{pmatrix}y(0)\\y(1)\\y(2)\end{pmatrix}$ with $x(0) = (2\,0)$, $u(0) = (0\,0)$ and $u(1) = (0\,2)$; so, it suffices to solve $\begin{pmatrix}CA^2 & CAB & CB & D\end{pmatrix}\begin{pmatrix}x(0)\\u(0)\\u(1)\\u(2)\end{pmatrix} = y(2)$

Then, we solve

$$Du(2) = y(2) - CA^2x(0) - CABu(0) - CBu(1)$$

So, $D^+ \begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}0 & 0\\2 & 0\end{pmatrix}\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$ is an approximate solution for $u(2)$.

Indeed, we can detect that there was a transmission error in the second part of the sequence $y(2) = (0\,2)$ as well, yet we approximate the original input sequence: $u(2) = (0\,0)$.

For this case, $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,1,\ 0\,2)$ there were several errors during transmission, on two sequences.

The decoded sequence is $u = (u(0), u(1), u(2)) = (0\,0,\ 0\,2,\ 0\,0)$, with initial condition: $x(0) = (2\,0)$

In fact, taking into account that there exists D^+, we can obtain an approximate solution with $x(0) = 0$ solving in the first step $D^+u(0) = y(0)$.

Example 4.3.3. We are going to see the decoding of words, with the code in which $p \geq k$. in the field $\mathbb{F}_7$, let (A_1, B_1, C_1, D_1) with

$$A = \begin{pmatrix} 1 & 3 \\ 4 & 1 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 2 \end{pmatrix},\ C = \begin{pmatrix} 5 & 2 \\ 0 & 6 \\ 3 & 0 \end{pmatrix},\ D = \begin{pmatrix} 2 \\ 5 \\ 6 \end{pmatrix} \tag{4.6}$$

We try to decode the sequence: $y = (y(0), y(1), y(2)) = (3\,1\,5,\ 6\,0\,2,\ 4\,2\,2)$

Observe that in this case, the system is not output-observable.

Step 1: We have $\ell = 2$

Step 2: We have

$$D = \begin{pmatrix} 2 \\ 5 \\ 6 \end{pmatrix};$$

then we solve $\begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$.

We choose to fix $x(0) = 0$. So, we solve

$$Du(0) = \begin{pmatrix} 2 \\ 5 \\ 6 \end{pmatrix} u(0) = y(0) = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix}.$$

The system is clearly incompatible, but the matrix D verifies conditions for existence of D^+ that in this particular case the pseudoinverse of the matrix D is given by $D^+ = (D^tD)^{-1}D^t$. So, $D^+ = \begin{pmatrix} 1 & 6 & 3 \end{pmatrix}$ Then,

$$D^+y(0) = \begin{pmatrix} 1 & 6 & 3 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 \end{pmatrix} = u(0)$$

We get $u(0) = 3$; when verifying $Du(0) = \begin{pmatrix} 6 \\ 1 \\ 4 \end{pmatrix}$. Here, at least we detect errors on two elements of the sequence. Indeed, we get for $x(0) = (0)$, $u(0) = (3)$.

Step 3: We solve $\widehat{T}_1 \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$ with $u(0) = 3$. So, it suffices to solve $\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = y(1) = \begin{pmatrix} 6 \\ 0 \\ 2 \end{pmatrix}$

Then, we solve

$$Du(1) = y(1) - CBu(0).$$

So, $D^+ \begin{pmatrix} 1 \\ 6 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 6 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 6 \\ 2 \end{pmatrix} = 1 = u(1)$. Then the solution $u(1)$ is (1).

When we verify, $\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 5 & 5 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 6 \end{pmatrix} \neq \begin{pmatrix} 6 \\ 0 \\ 2 \end{pmatrix}$.

So, we detected errors in the second sequence, and our approximate solution $u(1)$ is 1.

Finally, we solve $\widehat{T}_2 \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix}$ with $u(0) = 3$, and $u(1) = 1$.

So, we solve $\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2) = \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix}$ with $u(0) = 3$, and $u(1) = 1$.

Then, we solve

$$Du(2) = y(2) - CABu(0) - CBu(1)$$

We get $Du(2) = \begin{pmatrix} 3 \\ 3 \\ 4 \end{pmatrix}$;

$$u(2) = D^{+}(y(2) - CABu(0) - CBu(1)) = \begin{pmatrix} 1 & 6 & 3 \end{pmatrix} \begin{pmatrix} 3 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 5 \end{pmatrix} = u(2).$$

The approximate solution $u(2)$ is (5). Verifying, we have:

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} = \begin{pmatrix} 6 & 4 & 2 \\ 5 & 5 & 5 \\ 4 & 0 & 6 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 0 \end{pmatrix}$$

However, $d_H((4,\, 3\,0), (4,\, 2\,2)) = 2$; so, we detected two errors, and we approached the solution the best way possible.

For this case, $(y(0), y(1), y(2)) = (3\,1\,5,\ 6\,0\,2,\ 4\,2\,2)$ there were multiple errors during transmission.

The decoded sequence is $u = (u(0), u(1), u(2)) = (3,\ 1,\ 5)$, with initial condition $x(0) = (0\,0)$.

Example 4.3.4. We are going to see the decoding of words, with the code defined in the field $\mathbb{F}_{2^2}$; we have (A, B, C, D) with

$$A = \begin{pmatrix} 1 & \alpha+1 \\ 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ C = \begin{pmatrix} 0 & 1 \\ 0 & \alpha \end{pmatrix},\ D = \begin{pmatrix} 0 & \alpha \\ \alpha+1 & 1 \end{pmatrix} \tag{4.7}$$

Suppose that we want to decode the sequence

$y = (y(0), y(1), y(2)) = (1\,\alpha,\ 0\,1,\ \alpha+1\,1)$

First of all we observe that the system is output-observable because of the matrix D has full row rank.

Step 1: We have $\ell = 2$

Step 2: We have

$$D = \begin{pmatrix} 0 & \alpha \\ \alpha+1 & 1 \end{pmatrix};$$

then, we solve $T_0 \begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = y(0)$

We decide to choose $x(0) = (0\ 0)$; so we solve

$$D\,u(0) = \begin{pmatrix} 0 & \alpha \\ \alpha+1 & 1 \end{pmatrix} u(0) = y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

There exists $D^{+} = D^{-1} = \begin{pmatrix} 1 & \alpha \\ \alpha+1 & 0 \end{pmatrix}$; then, it follows

$u(0) = D^{-1}y(0) = \begin{pmatrix} 1 & \alpha \\ \alpha+1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha+1 \end{pmatrix}$.

Obtaining $u(0) = (1\ \alpha+1)$

Step 3: We solve $T_1 \begin{pmatrix} x(0) \\ u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \end{pmatrix}$ with $x(0) = (0\,0)$ and $u(0) = (1\,\alpha+1)$

Then, it suffices to solve

$$Du(1) = y(1) - CBu(0), \quad \text{with } u(0) = (1\ \alpha+1)$$

We get $D\,u(1) = y(1) - \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha+1 \end{pmatrix}$.

It follows $u(1) = D^{-1}\begin{pmatrix} 1 \\ \alpha+1 \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ \alpha+1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ \alpha+1 \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha+1 \end{pmatrix}$.

We get $u(1) = (0\ \alpha+1)$

Finally, we solve $T_2 \begin{pmatrix} x(0) \\ u(0) \\ u(1) \\ u(2) \end{pmatrix} = \begin{pmatrix} y(0) \\ y(1) \\ y(2) \end{pmatrix}$ with $x(0) = (0\,0)$, $u(0) = (1\,\alpha+1)$ and $u(1) = (0\,\alpha+1)$
Then, it suffices to solve

$Du(2) = y(2) - CABu(0) - CBu(1)$ with $u(0) = (1\,\alpha{+}1)$, and $u(1) = (0\ \alpha{+}1)$.

It follows $u(2) = D^{-1}\begin{pmatrix} \alpha+1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ \alpha+1 & 0 \end{pmatrix}\begin{pmatrix} \alpha+1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha \end{pmatrix}$.

We get $u(2) = (1\ \alpha)$.

For this case, $y = (y(0), y(1), y(2)) = (1\,\alpha,\ 0\,1,\ \alpha+1\ 1)$; there was no error during transmission;

With $x(0) = 0$, the decoded sequence is:

$u = (u(0), u(1), u(2)) = (1\,\alpha+1,\ 0\,\alpha+1,\ 1\,\alpha)$

Generalizing the process presented in the previous examples, we have the following result.

Proposition 4.3.1. *Let (A, B, C, D) be a representation of a convolutional code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$ the decoded sequence $u(0), u(1), \ldots, u(\ell)$ is obtained recursively as follows*

a) Choosing $x(0) = 0$

i) $u(0) = X_D y(0)$

ii) $u(1) = X_D(y(1) - CBu(0))$ *with $u(0)$ obtained in i)*

$\vdots$

ℓ) $u(\ell) = X_D(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1))$ *with $u(0)$, $u(1)$, ..., $u(\ell-1)$ obtained in i), ii), ..., $\ell - 1$).*

b) Not choosing $x(0)$

i) $\begin{pmatrix} x(0) \\ u(0) \end{pmatrix} = X_{(C\ D)} y(0)$

ii) $u(1) = X_D(y(1) - CAx(0) - CBu(0))$ *with $x(0)$, $u(0)$ obtained in i)*

$\vdots$

ℓ) $u(\ell) = X_D(y(\ell) - CA^{\ell}x(0) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$ *with $x(0)$, $u(0)$, $u(1)$, ..., $u(\ell-1)$ obtained in i), ii), ..., $\ell - 1$).*

Remark 13. This method is quite efficient for error detection; indeed, we can tell when there was a mistake within a sequence, by computing $d_H(y, \bar{y})$, y the output obtained from the approximate solution; however the correction rate is harder to figure out, since we only detect when a mistake occurs, and we assume the solution we get is the closest without any verification.

Remark 14. In case $p \leq k$, the correction process becomes heavy and its capacity decreases. Indeed, we have little information on the solution of the system we are trying to solve; therefore, the approximation of the solution may not be completely accurate.

Remark 15. In case $p \leq k$, the distance between the received sequence and the decoded one is most likely higher, than in the case when $p \geq k$. Indeed, the codeword space is less dense.

4.4 Second iterative decoding algorithm

The second algorithm for solving focuses more on detecting the error at first, before getting into the correction process. Indeed, we will consider that we need first to check wether or not the received sequence is the encoded one, and has not been compromised or modified due through the sending process. If so, we will correct the error first, and later figure out or deduce the original input message that is supposed to have been encoded. We will do so considering initial conditions, and output-observability matrix.

The objective of this second algorithm of decoding will be to try to approach at first, the word received from the encoding machine to the list of codewords.

Step 1: Set the initial conditions $x(0)$.

Step 2: With D we compute the list of u_0 codewords.

Then iteratively, generate the list of codewords, with matrix $T_\ell, \ell = 1, \ldots, l$ and store them in a set.

Step 3: Compute the distance between the received $y(0)$ and the set of $y(0)$ in the codewords. Compute d_H until it is minimal; then settle for the closest codeword $y(0)$ in the list; thus, the system (4.3) becomes solvable; deduce the corresponding input $u(0)$.

Iteratively, compute the distance between the received $y(1) \ldots y(\ell)$ and the set of corresponding codewords. Detect the minimum distance d_H between them and settle for the closest codeword $y(i), i \in 1, \ldots, \ell$ in the list; thus, the system (4.3) becomes compatible; deduce the corresponding input sequence.

Example 4.4.1. In $\mathbb{F}_2$, let us consider the code $\mathcal{C}(A, B, C, D)$ defined by:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \; C = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

We observe that the system is output-observable.

Suppose that we want to decode the sequence

$$y = (y(0), y(1), y(2)) = (1\,0,\ 0\,1,\ 0\,1)$$

Step 1: We set $x(0) = (0\,0)$

Step 2: We have

$$D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix};$$

that it has full row rank, so $(1\,0)$ belongs to $\operatorname{Im} D = \{(0\,0), (1\,0), (0\,1), (1\,1)\}$.

Then, we solve the system $Du(0) = y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, so, we obtain $u(0) = (0\,1)$.

Step 3: We have

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = y(1)$$

Then, we solve

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ u(1) \end{pmatrix} = y(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$(1\,0\,0\,1)$ belongs to $\operatorname{Im} \begin{pmatrix} D & \\ CB & D \end{pmatrix}$; we get: $u(1) = (1\,0)$.

Lastly, we have:

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2)$$

Then, we solve

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ u(2) \end{pmatrix} = y(2) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$(1\,0\,0\,1\,0\,1)$ belongs to $\operatorname{Im} \begin{pmatrix} D & & \\ CB & D & \\ CAB & CB & D \end{pmatrix}$; we finally get: $u(2) = (1\,0)$

For this case, $y = (y(0), y(1), y(2)) = (1\,0\ 0\,1\ 0\,1)$ there was no error during transmission.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (0\,1\ 1\,0\ 1\,0)$

Example 4.4.2. We consider that the encoding is done on $\mathbb{F}_3$, and the the code $\mathcal{C}(A, B, C, D)$ given by:

$$A = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}.$$

We observe that the system is not output-observable.

When we try to decode the sequence $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,1,\ 0\,2)$, the steps we follow are the following

Step 1: We set $x(0) = (0\,0)$

Step 2: We have

$$D = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix};$$

then we try to solve the system $Du(0) = y(0) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

We observe that $(2\,1)$ does not belong to $\operatorname{Im} D = \{(0\,0),\ (1\,0),\ (2\,0)\}$; however, $d_H((2\,1), (2\,0)) = 1$, so we can observe that there was an error on the second piece of the sequence, and changing $(2\,1)$ by $(2\,0)$, we have that $u(0)$ is either: $(0\,1)$ or $(\alpha\,1)$, $\forall \alpha \in \mathbb{F}_3$; we will consider $u(0) = (1\,1)$.

Step 3: We have

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = y(1)$$

$$Du(1) = y(1) - CBu(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

Then, we solve

$$Du(1) = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

$(0\,2)$ does not belong to $\operatorname{Im} D = \{(0\,0), (1\,0), (2\,0)\}$; however, $d_H((0\,0), (0\,2)) = 1$, which means that the error is in the second piece of the sequence, and

changing $y(1)$ by $\begin{pmatrix}0\\2\end{pmatrix}$ and solving the system we obtain that $u(1)$ is either $(1\,0)$, $(2\,0)$ or $(0\,0)$ in $\operatorname{Ker} D$; let us consider $u(1) = (1\,0)$.

Lastly, we have:

$$\begin{pmatrix}CAB & CB & D\end{pmatrix}\begin{pmatrix}u(0)\\u(1)\\u(2)\end{pmatrix} = y(2).$$

Then, we solve

$$Du(2) = y(2) - CABu(0) - CBu(1) = \begin{pmatrix}0\\2\end{pmatrix} \in \operatorname{Im} D,$$

the system is compatible with $u(2) \in \{(0\,1), (\alpha\,1)\}$, with $\alpha \in \mathbb{F}_3$. We decide to settle with $u(2) = (0\,1)$.

For this case, $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,1,\ 0\,2)$ there are multiple errors during transmission; there was 2 errors in the first 2 sequences. The original encoded sequence is: $y = (y(0), y(1), y(2)) = (2\,0\ 0\,2\ 0\,2)$.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (1\,1,\ 1\,0,\ 2\,0)$.

Example 4.4.3. In the field $\mathbb{F}_7$, we consider (A, B, C, D) with

$$A = \begin{pmatrix}1 & 3\\4 & 1\end{pmatrix},\ B = \begin{pmatrix}0\\2\end{pmatrix},\ C = \begin{pmatrix}5 & 2\\0 & 6\\3 & 0\end{pmatrix},\ D = \begin{pmatrix}2\\5\\6\end{pmatrix}.$$

We observe that the code is not output-observable.

We try to decode the sequence: $y = (y(0), y(1), y(2)) = (3\,1\,5,\ 6\,0\,2,\ 4\,2\,2)$

Step 1: We set $x(0) = (0\,0)$

Step 2: We can observe that D has full column rank; However, $y_0 = (3\,1\,5)$ does not belong to $\operatorname{Im} D = \{(0\,0\,0), (2\,5\,6), (4\,3\,5), (6\,1\,4), (1\,6\,3), (3\,4\,2), (5\,2\,1)\}$. So, the minimum distance is obtained over $(4\,3\,5)$, $(6\,1\,4)$ and $(3\,4\,2)$:

$$d_H((3\,1\,5), (4\,3\,5)) = d_H((3\,1\,5), (6\,1\,4)) = d_H((3\,1\,5), (3\,4\,2)) = 2;$$

which means that we have multiple choices. We decide to choose the second closest codeword in the list of codewords, that is to say, we consider the codeword $y(0) = (6\,1\,4)$ and then $u(0) = (3)$.

Step 3: We have

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = y(1)$$

So, we solve $Du(1) = \begin{pmatrix} 6 \\ 0 \\ 2 \end{pmatrix} - 3 \begin{pmatrix} 4 \\ 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \\ 2 \end{pmatrix}$

$(6\,0\,2)$ does not belong to $\mathrm{Im}D$; however, $d_H((1\,6\,2),(1\,6\,3)) = 1$, which means that the error is in the third element of the second sequence, and $u(1)$ is (4).

Lastly, we have:

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2),$$

$$Du(2) = \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix} - 3 \begin{pmatrix} 6 \\ 5 \\ 4 \end{pmatrix} - 4 \begin{pmatrix} 4 \\ 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}.$$

$(5\,2\,4)$ does not belong to $\mathrm{Im}\, D$. However, $d_H((5\,2\,4),(5\,2\,1)) = 1$, and the system is compatible with $u(2) = (6)$. We decide to settle with $u(2) = (6)$.

For this case, $y = (y(0), y(1), y(2)) = (3\,1\,5,\ 6\,0\,2,\ 4\,2\,2)$ there are errors during all transmission; there are 4 errors; 2 during the first transfer of information and 1 for the second and last transfer $d_H = 4$. The original encoded sequence is $y = (y(0), y(1), y(2)) = (6\,1\,4,\ 6\,0\,3,\ 4\,2\,6)$.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (3,\ 4,\ 6)$.

Example 4.4.4. We are going to see the decoding of words, with the code defined in the field $\mathbb{F}_{2^2}$; we have (A, B, C, D) with

$$A = \begin{pmatrix} 1 & \alpha+1 \\ 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ C = \begin{pmatrix} 0 & 1 \\ 0 & \alpha \end{pmatrix},\ D = \begin{pmatrix} 0 & \alpha \\ \alpha+1 & 1 \end{pmatrix}$$

(The same code as in (4.7))

We observe that the system is output-observable, since $\mathrm{rank}\, D$ is maximal in $\mathbb{F}_4$.

Suppose that we want to decode the sequence $y = (y(0), y(1), y(2)) = (1\ \alpha,\ \ 0\ 1,\ \ \alpha+1\ 1)$.

Step 1: We set $x(0) = (0\,0)$

Step 2: We have

$$D = \begin{pmatrix} 0 & \alpha \\ \alpha+1 & 1 \end{pmatrix};$$

since D has full row rank, so $(1\,\alpha)$ belongs to $\operatorname{Im} D = \{(0\ \alpha+1), (\alpha\ 1)\}$.

Then, we solve the system $Du(0) = y(0) = \begin{pmatrix} 1 \\ \alpha \end{pmatrix}$, so, we obtain $u(0) = (\alpha\ \alpha+1)$

Step 3: We solve

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = y(1).$$

We get $Du(1) = \begin{pmatrix} \alpha \\ \alpha \end{pmatrix}$. The vector $(\alpha\ \alpha)$ belongs to $\operatorname{Im}\begin{pmatrix} D \end{pmatrix}$; we get: $u(1) = (1\ 1)$

Lastly, we have

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2).$$

Then, we solve

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & \alpha \\ 0 & 0 & \alpha & 0 & \alpha+1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha+1 \\ 1 \\ 1 \\ u(2) \end{pmatrix} = \bar{y}(2) = \begin{pmatrix} \alpha+1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} \alpha \\ 1+\alpha \end{pmatrix}.$$

We have $Du(2) = \begin{pmatrix} \alpha \\ \alpha+1 \end{pmatrix}$. The vector $(\alpha\ \alpha+1)$ belongs to $\operatorname{Im} D$; we finally get $u(2) = (\alpha+1\ 1)$.

For this case, $y = (y(0), y(1), y(2)) = (1\ \alpha,\ \ 0\ 1,\ \ \alpha+1\ 1)$ there was no error during transmission;

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (\alpha\ \alpha+1,\ \ 1\ 1,\ \ \alpha+1\ 1)$

Generalising the process presented, we have the following result:

Proposition 4.4.1. *Let (A, B, C, D) be a representation of a convolutional code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$ the decoded sequence $u(0), u(1), \ldots, u(\ell)$ is obtained recursively as follows*

a) Setting $x(0) = 0$

b) i) Computing $d_H(y(0), \operatorname{Im} D)$;
if $d_H(y(0), \operatorname{Im} D) = 0$, we solve $Du(0) = y(0)$;
else, for some $y \in \operatorname{Im} D$ such that $d_H(y(0), \operatorname{Im} D) = \min d_H$, we solve $u(0)$ is a solution of: $Du(0) = y$.

ii) Computing $d_H(y(1) - CBu(0), \operatorname{Im} D)$;
if $d_H(y(1) - CBu(0), \operatorname{Im} D) = 0$, we solve $Du(1) = y(1) - CBu(0)$, with $u(0)$ obtained in i);
else, for some $y \in \operatorname{Im} D$ such that $d_H(y(1) - CBu(0), \operatorname{Im} D) = \min d_H$, we solve $Du(1) = y$, with $u(0)$ obtained in i)

$\vdots$

ℓ) Computing
$d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D)$;
if $d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = 0$, we solve $Du(l) = (y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$ with $u(0)$, $u(1)$, ..., $u(\ell-1)$ obtained in i), ii), ..., $\ell - 1)$;
else, for some $y \in \operatorname{Im} D$ such that
$d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = \min d_H$, we solve $Du(l) = y$, with $u(0)$, $u(1)$, ..., $u(\ell-1)$ obtained in i), ii), ..., $\ell - 1)$

Corollary 4.4.1. *Let (A, B, C, D) be a representation of a convolutional code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. If D has full column rank and $n \leq 2k$, given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $u(0), u(1), \ldots, u(\ell)$ is obtained recursively as follows:*

a) Setting $x(0) = 0$

b) i) $u(0)$ is a solution of $Du(0) = y(0)$.

ii) $u(1)$ *is a solution of* $Du(1) = y(1) - CBu(0)$*, with* $u(0)$ *obtained in i);*

$\vdots$

$\ell)$ $u(\ell)$ *is a solution of*
$Du(l) = (y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1))$
with $u(0)$*,* $u(1)$*, ...,* $u(\ell-1)$ *obtained in i), ii), ...,* $\ell-1$*), with* $u(0)$*,* $u(1)$*, ...,* $u(\ell-1)$ *obtained in i), ii), ...,* $\ell-1$*)*

Proof. If D has full column rank, and $n \leq 2k$, then $p = k$, and the matrix D is invertible, so, for all $y(0)$, there exists $u(0)$ such that $Du(0) = y(0)$. Similarly, for all $i \in \{1, \ldots, \ell\}$ and for all $y(i)$, there exists $u(i)$ such that $Du(i) = y(i) - CA^{i-1}Bu(0) - CA^{i-2}Bu(1) - \ldots - CBu(i-1)$ □

Remark 16. This decoding is both a detection and correction method; at first, we detect the error; and then we try to correct.

Remark 17. The resolution with this method depends heavily on the matrix D.

Remark 18. Suppose that we have the code $\mathcal{C}$, that is not output-observable. Then, we know that D does not have full row rank, and it can potentially increase the decoding time.

In order to get into the decoding of concatenated convolutional codes, we will be working within two cases:

Case 1 The first method consists of considering that the mistake occurred somewhere in between the encodings, before the concatenation, but not after. Therefore, we would have to first of all try to recover the original input, and then attempt to detect where the error occurred. For this decoding algorithm, we will be trying to assess the value of each and every code's capacity to decode, one at the time.

Case 2 The second angle would be to consider that in case of error, it occurred after the concatenation, which is after encoding with the second code $\mathcal{C}_2$. The approach for this decoding method consists of detecting the error, correct it and then recover the encoded each part coming from every convolutional code.

4.4.1 Iterative decoding algorithm for serial concatenated codes

For this decoding algorithm, we will try to assess the value of each and every code's capacity to decode, one at the time.

Let (A, B, C, D) be the serial concatenation of the convolutional codes (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) over $\mathbb{F}_q$ with $q = q_1^m$ and q_1 prime.

It is possible to decode this system decoding both systems in the following manner.

Proposition 4.4.2. *In this conditions, the decoded sequence* $u = (u(0), u(1), \ldots, u(\ell))$ *of the sequence* $y = (y(0), y(1), \ldots, y(\ell)$ *is given by*

$$X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y$$

and

$$X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y$$

is the decoded sequence of the y *with respect the second system.*

Proof. It suffices to observe that $u_2(t) = y_1(t)$ and

$$\begin{aligned} X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y &= u_2 = y_1 \\ X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} y_1 &= u. \end{aligned}$$

□

Let us look at more examples of decoding of concatenated convolutional codes, with decoding for each an every code intervening in the concatenation process. We need to dissociate the decoding of both convolutional codes, and separate them into two, and recover the initial sequence before each encoding.

Example 4.4.5. For this case, we look at a serial concatenated model.

Over the field $\mathbb{F}_5$, we consider a concatenated serial code $\mathcal{C}$ of a $(3, 2, 1)$-convolutional code $\mathcal{C}_1$, and a $(3, 1, 2)$-convolutional code $\mathcal{C}_2$, where

$$A_1 = \begin{pmatrix} 0 \end{pmatrix}, \; B_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \; C_1 = \begin{pmatrix} 0 \end{pmatrix}, \; D_1 = \begin{pmatrix} 1 & 3 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}, \; B_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \; C_2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \; D_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

The serial concatenation of those two codes is

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 1 & 3 \\ 0 & 0 \end{pmatrix}, C = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}, D = \begin{pmatrix} 3 & 4 \\ 0 & 0 \end{pmatrix}$$

We will try to decode the sequence $y = (2\,1\ 3\,2\ 1\,1)$. We will consider the encoded sequence with the $\mathcal{C}_1$-code $y_1 = (y_1(0), y_1(1), y_1(2))$, and the one with the $\mathcal{C}_2$-code $y_2 = (y_2(0), y_2(1), y_2(2))$.

Step 1: we fix $x(0) = (x_1(0), x_2(0)) = (0\,0)$

Step 2: we can observe that $p = k$ and D does not have full row rank; indeed, $y_0 = (2\,1)$ does not belong to ImD. The distance $d_H((2\,1), (2\,0)) = 1$, which means an error during transmission has affected the second piece of the sequence, and $u(0)$ is either $(1\,1)$, $(2\,4)$, $(3\,2)$, $(4\,0)$, or $(0\,3)$; we consider $u(0) = (1\,1)$; $u(0)$ is the original input, the message to be encoded.

In this case, because it is the serial concatenation, we have that $D = D_2 D_1$. Then, we can deduce $y(0) = Du(0) = D_2 D_1 u(0)$, which means that $y_1(0) = D_1 u(0) = \begin{pmatrix} 1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = (4)$.
As well as for the second encoder, we can deduce $y(0) = Du(0) = D_2 y_1(0)$, which means that $y_2(0) = D_2 y_1(0) = \begin{pmatrix} 3 \\ 0 \end{pmatrix} (4) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$.

In conclusion, we get $y_1(0) = (4)$ and $y_2(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$

Step 3: We have

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(1) \end{pmatrix}.$$

Then, we solve

$$\begin{pmatrix} 1 & 3 & 3 & 4 \\ 2 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ u(1) \end{pmatrix} = \begin{pmatrix} y(1) \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}.$$

Then, solving that system comes down to solving $Du(1) = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$ $(4\,4)$ does not belong to Im D; however, $d_H((4\,4), (4\,0)) = 1$; then, a solution for $Du(1) = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$ is $u(1) = (0\,1)$

Then, we can deduce $y(1) = Du(1) + CBu(0) = D_2D_1u(1) + CBu(0)$, which means that $y_1(1) = D_1u(1) = \begin{pmatrix}1 & 3\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}3\end{pmatrix}$.
As well as for the second encoder, we can deduce $Du(1) = D_2y_1(1) = y_2(1)$, which means that
$y_2(1) = D_2y_1(1) + CBu(0) = \begin{pmatrix}3\\0\end{pmatrix}\begin{pmatrix}3\end{pmatrix} + \begin{pmatrix}4\\3\end{pmatrix} = \begin{pmatrix}4\\0\end{pmatrix} + \begin{pmatrix}4\\3\end{pmatrix} = \begin{pmatrix}3\\3\end{pmatrix}$.
In conclusion, we get: $y_1(1) = \begin{pmatrix}3\end{pmatrix}$ and $y_2(1) = \begin{pmatrix}3\\0\end{pmatrix}$.

Lastly, we have

$$\begin{pmatrix}CAB & CB & D\end{pmatrix}\begin{pmatrix}u(0)\\u(1)\\u(2)\end{pmatrix} = y(2).$$

Then, we solve

$$\begin{pmatrix}0 & 0 & 1 & 3 & 3 & 4\\1 & 3 & 2 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\1\\0\\1\\u(2)\end{pmatrix} = y(2) = \begin{pmatrix}1\\1\end{pmatrix}.$$

Then, solving that system comes down to solving $Du(2) = \begin{pmatrix}3\\1\end{pmatrix}(3\,1)$ does not belong to $\operatorname{Im} D$. However, $d_H((3\,1),(3\,0)) = 1$, and the system is compatible with $u(2) = (1\,0)$. We decide to settle with $u(2) = (1\,0)$.

Then, we can deduce $y(2) = Du(2) + CBu(1) + CABu(0) = D_2D_1u(2) + CBu(1) + CABu(0)$, which means that $y_1(2) = D_1u(2) = \begin{pmatrix}1 & 3\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = (1)$
As well as for the second encoder, we can deduce $y_2(2) = Du(2) + CBu(1) + CABu(0) = D_2y_1(2) + CBu(1) + CABu(0)$, which means that $y_2(2) = \begin{pmatrix}3\\0\end{pmatrix}(1) + \begin{pmatrix}3\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$.

In conclusion, we get $y_1(2) = \begin{pmatrix}1\end{pmatrix}$ and $y_2(2) = \begin{pmatrix}1\\0\end{pmatrix}$.

For this case, $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,2,\ 1\,1)$ there was errors during all transmission; there was 3 errors: $d_H = 3$. The original encoded sequence was: $y = (y(0), y(1), y(2)) = (2\,0\ 3\,3\ 1\,0)$.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (1\,1,\ 0\,1,\ 1\,0)$.

Generalizing the process presented in those previous examples, we have the following results.

Case 1 If we consider the first case, where the errors could occur in between concatenation while encoding, we have the following proposition.

Proposition 4.4.3. *Let (A, B, C, D) be a representation of a serial concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ as the inner code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $u_2(0), u_2(1), \ldots, u_2(\ell)$ input of the inner code $\mathcal{C}_2$ is obtained recursively as follows*

a) Choosing $x(0) = 0$

i) $u_2(0) = X_{D_2} y(0)$

ii) $u_2(1) = X_{D_2}(y(1) - CBX_{D_1} u_2(0))$ *with $u_2(0)$ obtained in i)*

$\vdots$

ℓ) $u_2(\ell) = X_{D_2}(y(\ell) - CA^{\ell-1}BX_{D_1} u_2(0) - CA^{\ell-2}BX_{D_1} u_2(1) - \ldots - CBX_{D_1} u_2(\ell-1))$ *with $u_2(0)$, $u_2(1)$, ..., $u_2(\ell-1)$ obtained in i), ii), ..., $\ell - 1$).*

b) Not Choosing $x(0)$

i) $u_2(0) = X_{D_2}(y(0) - Cx(0))$

ii) $u_2(1) = X_{D_2}(y(1) - CAx(0) - CBX_{D_1} u_2(0))$ *with $x(0)$, $u_2(0)$ obtained in i)*

$\vdots$

ℓ) $u_2(\ell) = X_{D_2}(y(\ell) - CA^{\ell}x(0) - CA^{\ell-1}BX_{D_1} u_2(0) - CA^{\ell-2}BX_{D_1} u_2(1) - \ldots - CBX_{D_1} u_2(\ell-1))$ *with $x(0)$, $u_2(0)$, $u_2(1)$, ..., $u_2(\ell-1)$ obtained in i), ii), ..., $\ell - 1$).*

From the precedent propositions on the serial concatenated decoding, we can give the following result.

Proposition 4.4.4. *Let (A, B, C, D) be a representation of a serial concatenated code of two convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ as the inner code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $u_1(0), u_1(1), \ldots, u_1(\ell)$ of the outer code $\mathcal{C}_1$ is obtained recursively as follows*

a) Choosing $x(0) = 0$

i) $u_1(0) = X_D y(0)$

ii) $u_1(1) = X_D(y(1) - CBu_1(0))$ *with* $u_1(0)$ *obtained in i)*

$\vdots$

ℓ) $u_1(\ell) = X_D(y(\ell) - CA^{\ell-1}Bu_1(0) - CA^{\ell-2}Bu_1(1) - \ldots - CBu_1(\ell-1))$ *with* $u_1(0)$, $u_1(1)$, …, $u_1(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$).

b) Choosing $x(0) \neq 0$

i) $u_1(0) = X_D(y(0) - Cx(0))$

ii) $u_1(1) = X_D(y(1) - CAx(0) - CBu_1(0))$ *with* $x(0)$, $u_1(0)$ *obtained in i)*

$\vdots$

ℓ) $u_1(\ell) = X_D(y(\ell) - CA^{\ell}x(0) - CA^{\ell-1}Bu_1(0) - CA^{\ell-2}Bu_1(1) - \ldots - CBu_1(\ell-1))$ *with* $x(0)$, $u_1(0)$, $u_1(1)$, …, $u_1(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$).

This last method is an alternative of the general decoding of the convolutional code, since recovering the input of the outer code is equivalent to recovering the input of the whole concatenated code.

Case 2

Proposition 4.4.5. *Let (A, B, C, D) be a representation of a serial concatenated code of two convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ as the inner code over a field $\mathbb{F}_q$ with $q = q_1^m$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $(u(0), u(1), \ldots u(\ell))$ input of the outer code $\mathcal{C}_1$ is obtained recursively as follows*

a) Choosing $x(0) = 0$

b) *i)* *Computing* $d_H(y(0), \operatorname{Im} D)$*;*
if $d_H(y(0), \operatorname{Im} D) = 0$, $u(0)$ *is a solution of:* $D.u(0) = y(0)$*;*
else, for some $y \in \operatorname{Im} D$ *such that* $d_H(y(0), y) = \min d_H(y(0), \operatorname{Im} D)$, $u(0)$ *is a solution of:* $D.u(0) = y$.

ii) *Computing* $d_H(y(1) - CBu(0), \operatorname{Im} D)$, *with* $u(0)$ *obtained in i);*
if $d_H(\, y(1) - CBu(0)\, , \operatorname{Im} D) = 0$, $u(1)$ *is a solution of:*
$D.u(1) = y(1) - CBu(0)$, *with* $u(0)$ *obtained in i);*
else, for some $y \in \operatorname{Im} D$ *such that* $d_H(\, y(1) - CBu(0)\, , y) = \min d_H(y(1) - CBu(0), \operatorname{Im} D)$,
$u(1)$ *is a solution of:* $Du(1) = y$, *with* $u(0)$ *obtained in i)*

$\vdots$

ℓ) *Computing* $d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell - 1), \operatorname{Im} D)$, *with* $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell-1$).
if $d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = 0$, $u(\ell)$ *is a solution of:*
$Du(l) = y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$ *with* $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$*);*
else, for some $y \in \operatorname{Im} D$ *such that*
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)\, , y) = \min d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)\, , \operatorname{Im} D)$,
$u(l)$ *is a solution of:* $Du(l) = y$, *with* $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$).

4.4.2 Iterative decoding algorithm for systematic serial concatenated codes

When it comes to the decoding in a systematic serial concatenation case, we need to keep in mind that the original input is encoded with D_1, and the result is sent over to the inner encoder. The classic decoding method only goes back to the very first input; with this algorithm we will be trying to decode every step of the way.

Let (A, B, C, D) be the serial concatenation of the convolutional codes (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) over $\mathbb{F}_q$ with $q = q_1^m$ and q_1 prime.

It is possible to decode this system decoding both systems in the following manner.

Proposition 4.4.6. *In this conditions, the decoded sequence* $u = (u(0), u(1), \ldots, u(\ell))$ *of the sequence*

$$y = (y_1, y_2) = ((y_1(0), y_2(0)), (y_1(1), y_2(1)), \ldots, (y_1(\ell), y_2(\ell))$$

is given by

$$X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} y_1 = X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y_2$$

and

$$X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y_2$$

is the decoded sequence of the y_2 *with respect the second system.*

Proof. It suffices to observe that $u(t) = X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} y_1(t)$;

and that in the conditions of the systematic serial concatenation $y_1(t) = X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} y_2(t)$. □

In similar conditions as mentioned above, it is possible to detect eventual errors occurred during encoding using the following process.

Proposition 4.4.7. *For* (A, B, C, D) *a representation of a systematic serial concatenated code of two convolutional codes* $\mathcal{C}_1(A_1, B_1, C_1, D_1)$*, as the outer code and* $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ *as the inner code over a field* $\mathbb{F}_q$ *with* $q = q_1^m$*,* q_1 *being prime, considering the output sequence* $y = (y(0), y(1), \ldots, y(\ell))$*,* $Q = (I_{m-k} \mid 0_{m-k \times n-(m-k)})$ *and* $R = (0_{n-(m-k)\times m-k} \mid I_{n-(m-k)})$*, the approximation of errors during encoding can be assessed by:*

$$w(\{X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} Qy\} - \{X_{\widehat{T}_\ell(A_1,B_1,C_1,D_1)} X_{\widehat{T}_\ell(A_2,B_2,C_2,D_2)} Ry\})$$

In case $\min w > 0$*, the number of errors is definitely not null.*

Example 4.4.6. For this case, we look at a systematic serial concatenated model.

We decode the input from the outer code $\mathcal{C}_1$.

In $\mathbb{F}_7$, we consider two convolutional codes $\mathcal{C}_1$ a $(3, 2, 1)$-convolutional code, and $\mathcal{C}_2$ a $(3, 1, 2)$-convolutional code, with respectively

$$A_1 = \begin{pmatrix} 0 \end{pmatrix},\ B_1 = \begin{pmatrix} 2 & 1 \end{pmatrix},\ C_1 = \begin{pmatrix} 3 \\ 5 \end{pmatrix},\ D_1 = \begin{pmatrix} 1 & 0 \\ 4 & 2 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 3 & 0 \\ 5 & 2 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, D_2 = \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix}$$

After the serial concatenation of those two codes, we get

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 4 \\ 4 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 2 & 1 \\ 3 & 0 \\ 6 & 4 \end{pmatrix}, C = \begin{pmatrix} 3 & 0 & 0 \\ 5 & 0 & 0 \\ 4 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 \\ 4 & 2 \\ 6 & 2 \\ 0 & 3 \end{pmatrix}$$

We will try to decode the sequence $y = (y(0), y(1), y(2)) = (2\,1\,1\,3,\ 3\,0\,2\,1,\ 1\,4\,1\,5)$ We will consider the encoded sequence with the $\mathcal{C}_1$-code $y_1 = (y_1(0), y_1(1), y_1(2))$, and the one with the $\mathcal{C}_2$-code $y_2 = (y_2(0), y_2(1), y_2(2))$.

Step 1: we fix $x(0) = (0\,0)$

Step 2: In order to recover $u(0)$, we decide to approach the supposed input sequence by computing the Moore-Penrose pseudoinverse if exists; otherwise a generalized inverse will do.

Taking into account that

$$\operatorname{rank}(D) = \operatorname{rank}(D^t) = \operatorname{rank}(DD^t) = \operatorname{rank}(D^tD) = 1$$

we have that D^+ exists and $D^+ = (D^tD)^{-1}D^t = \begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 1 & 4 & 6 & 0 \\ 0 & 2 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 6 & 0 & 5 & 6 \\ 2 & 3 & 0 & 3 \end{pmatrix}$.

Then, $u(0) = D^+y(0) = \begin{pmatrix} 6 & 0 & 5 & 6 \\ 2 & 3 & 0 & 3 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

All possible solutions are $u(0) + \begin{pmatrix} x \\ y \end{pmatrix} - D^+D\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

Step 3: we have

$$\begin{pmatrix} CB & D \end{pmatrix}\begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(1) \end{pmatrix}.$$

Then, solving that system comes down to solving $Du(1) = \begin{pmatrix} 4 \\ 4 \\ 1 \\ 0 \end{pmatrix}$,

$$\begin{pmatrix} 6 & 0 & 5 & 6 \\ 2 & 3 & 0 & 3 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} u(1) \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \end{pmatrix}.$$

All possible solutions are $u(1) + \begin{pmatrix} x \\ y \end{pmatrix} - D^+ D \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \end{pmatrix}$.

Then, we have a solution $u(1) = (1\,6)$.

Lastly, we have

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = \begin{pmatrix} y(2) \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 1 \\ 5 \end{pmatrix}$$

Then, we solve $u(2) = D^+(y(2) - CABu(0) - CBu(1))$;

$$\begin{pmatrix} 0 & 0 & 6 & 3 & 1 & 0 \\ 0 & 0 & 3 & 5 & 4 & 2 \\ 0 & 4 & 4 & 4 & 6 & 2 \\ 4 & 5 & 5 & 4 & 0 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 1 \\ 6 \\ u(2) \end{pmatrix} = y(2) = \begin{pmatrix} 1 \\ 4 \\ 1 \\ 5 \end{pmatrix}$$

This comes down to solving $Du(2) = \begin{pmatrix} 1 \\ 4 \\ 1 \\ 5 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 3 \\ 5 \\ 0 \\ 1 \end{pmatrix}$. We already have $D^+ = \begin{pmatrix} 6 & 0 & 5 & 6 \\ 2 & 3 & 0 & 3 \end{pmatrix}$; then $u(2) = D^+ \begin{pmatrix} 5 \\ 6 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$.

All possible solutions are $u(2) + \begin{pmatrix} x \\ y \end{pmatrix} - D^+ D \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$.

We have $u(2) = (1\,3)$.

We finally get, after decoding of the outer code $u = (u(0), u(1), u(2)) = (0\,2\;\;1\,6\;\;1\,3)$.

For this same case of systematic serial concatenation, after decoding, we will try to approximate the number of eventual errors occurred during the transmission.

At step 0
let us compute: $d_H(X_{D_2} Ry(0), D_1 u(0))$;
we have $D_2 = D_2^t = \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix}$; therefore

$$\text{rank}(D_2) = \text{rank}(D_2^t) = \text{rank}(D_2 D_2^t) = \text{rank}(D_2^t D_2) = 2;$$

then

$$D_2^{-1} = \begin{pmatrix} 6 & 3 \\ 3 & 1 \end{pmatrix}$$

$X_{D_2} Ry(0) = D_2^{-1} Ry(0) = \begin{pmatrix} 6 & 3 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \end{pmatrix}$
$D_1 u(0) = \begin{pmatrix} 1 & 0 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$
which means that: $d_H(X_{D_2} Ry(0), D_1 u(0)) = d_H(D_2^{-1} Ry(0), D_1 u(0)) = d_H\left(\begin{pmatrix} 1 \\ 6 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \end{pmatrix}\right) = 2$.

At step 1
let us compute: $d_H(D_2^{-1} R(y(1) - CBu(0)), D_1 u(1))$;
$D_2^{-1} R(y(1) - CBu(0)) = \begin{pmatrix} 6 & 3 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$
$D_1 u(1) = \begin{pmatrix} 1 & 0 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
which means that:
$d_H(D_2^{-1} R(y(1) - CBu(0)), D_1 u(1)) = d_H\left(\begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}\right) = 2$.

At step 2
let us compute: $d_H(D_2^{-1} R(y(2) - CABu(0) - CBu(1)), D_1 u(2))$;

$$D_2^{-1}R(y(2) - CABu(0) - CBu(1)) = \begin{pmatrix} 6 & 3 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$D_1 u(2) = \begin{pmatrix} 1 & 0 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

which means that:

$$d_H(D_2^{-1}R(y(2) - CABu(0) - CBu(1)), D_1 u(2)) = d_H\left(\begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}\right) = 2.$$

Finally, after decoding the sequence, we can conclude that in between both encoding, the sequence was subject to errors all the way; which means that the sequence received was distorted.

Case 1
Generalizing the previous example, we can generalize the process.

Proposition 4.4.8. *Let (A, B, C, D) be a representation of a systematic serial concatenated code of two convolutional codes $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, an (m, k, δ_1)-code as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, an $(n, m-k, \delta_2)$-code as the inner code over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime.*
Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $(u(0), u(1), \ldots, u(\ell))$ input of the outer code $\mathcal{C}_1$ is obtained recursively as follows:

a) Choosing $x(0) = 0$

i) $u(0) = X_D y(0)$

ii) $u(1) = X_D(\, y(1) - CBu(0)\,)$ *with $u(0)$ obtained in i)*

$\vdots$

ℓ) $u(\ell) = X_D(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)\,)$ *with $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$).*

b) Choosing $x(0) \neq 0$

i) $u(0) = X_D(y(0) - Cx(0))$

ii) $u(1) = X_D(\, y(1) - CAx(0) - CBu(0)\,)$ *with $x(0)$, $u(0)$ obtained in i)*

$\vdots$

ℓ) $u(\ell) = X_D(\, y(\ell) - CA^{\ell}x(0) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)\,)$ *with $x(0)$, $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$).*

The methods of decoding stated earlier allow us to directly recover inputs of each encoder. Let us not forget that in this case, we considered that if errors happened, they did in between the two encodings.

In order to evaluate the errors that happened, we will be proceeding as follows:

Proposition 4.4.9. *Let (A, B, C, D) be a representation of a systematic serial concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, an (m, k, δ_1)-code as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, an $(n, m-k, \delta_2)$-code as the inner code over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime.*
Given the decoded sequence $(u(0), u(1), \ldots, u(\ell))$ input of the outer code $\mathcal{C}_1$ and the sequence $y = (y(0), y(1), \ldots, y(\ell))$, and considering $Q = (I_{m-k} \mid 0_{m-k\times n-(m-k)})$ and $R = (0_{n-(m-k)\times m-k} \mid I_{n-(m-k)})$, the approximative number of errors is obtained recursively as follows:

a) Choosing $x(0) = 0$

i) $\min d_H(\{X_{D_2} R y(0)\}, D_1 u(0))$, *at step* 0

ii) $\min d_H(\{X_{D_2} R(\ y(1) - CBu(0))\}, D_1 u(1))$, *at step* 1

$\vdots$

ℓ) $\min d_H(\{X_{D_2} R(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell - 1))\}, D_1 u(\ell))$, *at step ℓ.*

b) Choosing $x(0) \neq 0$

i) $\min d_H(\{X_{D_2} R(y(0) - Cx(0))\}, D_1 u(0))$, *at step* 0

ii) $\min d_H(\{X_{D_2} R(\ y(1) - CAx(0) - CBu(0))\}, D_1 u(1))$, *at step* 1

$\vdots$

ℓ) $\min d_H(\{X_{D_2} R(y(\ell) - CA^{\ell}x(0) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell - 1))\}, D_1 u(\ell))$, *at step ℓ.*

Proof. It suffices to realize that for the systematic serial concatenated code, the matrix D is such that:

$$D = \begin{pmatrix} D_1 \\ D_2 D_1 \end{pmatrix}$$

□

In order to give the decoding of the inner code, we will generalize from the previous examples

Proposition 4.4.10. *Let (A, B, C, D) be a representation of a systematic serial concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, an (m, k, δ_1)-code as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, an $(n, m-k, \delta_2)$-code as the inner code over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, and considering $Q = (0_{n-(m-k)\times m-k} \mid I_{n-(m-k)})$, the decoded sequence $(u_2(0), u_2(1), \ldots, u_2(\ell))$ input of the inner code $\mathcal{C}_2$ is obtained recursively as follows:*

a) Choosing $x(0) = 0$

i) $u_2(0) = X_{D_2} Q y(0)$

ii) $u_2(1) = X_{D_2} Q(y(1) - CBX_{D_1} u_2(0))$ *with $u_2(0)$ obtained in i)*

$\vdots$

ℓ) $u_2(\ell) = X_{D_2} Q(\, y(\ell) - CA^{\ell-1} BX_{D_1} u_2(0) - CA^{\ell-2} BX_{D_1} u_2(1) - \ldots - CBX_{D_1} u_2(\ell-1)\,)$ *with $u_2(0)$, $u_2(1)$, …, $u_2(\ell-1)$ obtained in i), ii), …, $\ell - 1$).*

b) Choosing $x(0) \neq 0$

i) $u_2(0) = X_{D_2} Q(y(0) - Cx(0))$

ii) $u_2(1) = X_{D_2} Q(y(1) - CAx(0) - CBX_{D_1} u_2(0))$ *with $x(0)$, $u_2(0)$ obtained in i)*

$\vdots$

ℓ) $u_2(\ell) = X_{D_2} Q(\, y(\ell) - CA^{\ell} x(0) - CA^{\ell-1} BX_{D_1} u_2(0) - CA^{\ell-2} BX_{D_1} u_2(1) - \ldots - CBX_{D_1} u_2(\ell-1)\,)$ *with $x(0)$, $u_2(0)$, $u_2(1)$, …, $u_2(\ell-1)$ obtained in i), ii), …, $\ell - 1$).*

In order to assess the eventual error occurred anywhere in between concatenation, we can proceed with the following result:

Case 2
Generalizing the previous examples, we can generalize the process:

Proposition 4.4.11. *Let (A, B, C, D) be a representation of a systematic serial concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, an*

(m, k, δ_1)-code as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, an $(n, m-k, \delta_2)$-code as the inner code over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $(u(0), u(1), \ldots, u(\ell))$ input of the outer code $\mathcal{C}_1$ is obtained recursively as follows:

a) Choosing $x(0) = 0$

b) i) Computing $d_H(y(0), \operatorname{Im} D)$;
if $d_H(y(0), \operatorname{Im} D) = 0$, $u(0)$ is a solution of: $D.u(0) = y(0)$;
else, for some $y \in \operatorname{Im} D$ such that $d_H(y(0), y) = \min d_H(y(0), \operatorname{Im} D)$, $u(0)$ is a solution of: $Du(0) = y$.

ii) Computing $d_H(y(1) - CBu(0), \operatorname{Im} D)$, with $u(0)$ obtained in i);
if $d_H(\, y(1) - CBu(0), \operatorname{Im} D) = 0$, $u(1)$ is a solution of:
$Du(1) = y(1) - CBu(0)$, with $u(0)$ obtained in i);
else, for some $y \in \operatorname{Im} D$ such that $d_H(\, y(1) - CBu(0), y) = \min d_H(y(1) - CBu(0), \operatorname{Im} D)$, $u(1)$ is a solution of: $Du(1) = y$, with $u(0)$ obtained in i)

$\vdots$

ℓ) Computing
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D_1\,)$, with $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$).
if
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = 0$, $u(\ell)$ is a solution of:
$D.u(l) = y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$ with $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$);
else, for some $y \in \operatorname{Im} D$ such that
$d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), y) = \min d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D)$, $u(l)$ is a solution of: $Du(l) = y$, with $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$).

In order to do the other decoding, we can use the following result.

Proposition 4.4.12. *Let (A, B, C, D) be a representation of a systematic serial concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, an (m, k, δ_1)-code as the outer code and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, an $(n, m-k, \delta_2)$-code as the inner code over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime. Given*

the sequence $y = (y(0), y(1), \ldots, y(\ell))$*, and considering* $Q = (0_{n-(m-k)\times m-k} \mid I_{n-(m-k)})$*, the decoded sequence* $(u_2(0), u_2(1), \ldots, u_2(\ell))$ *input of the inner code* $\mathcal{C}_2$ *is obtained recursively as follows:*

a) Choosing $x(0) = 0$

b) i) Computing $d_H(Qy(0), \operatorname{Im} D_2)$*;*

if $d_H(Qy(0), \operatorname{Im} D_2) = 0$*,* $u_2(0)$ *is a solution of:* $D_2.u_2(0) = Qy(0)$*; else, for some* $y \in \operatorname{Im} D_2$ *such that* $d_H(Qy(0), y) = \min d_H(Qy(0), \operatorname{Im} D_2)$*,* $u_2(0)$ *is a solution of:* $D_2u_2(0) = y$*.*

ii) Computing $d_H(Q(y(1) - CBX_{D_1}u_2(0)), \operatorname{Im} D_2)$*, with* $u_2(0)$ *obtained in i);*

if $d_H(Q(y(1) - CBX_{D_1}u_2(0)), \operatorname{Im} D_2) = 0$*,* $u_2(1)$ *is a solution of:* $D_2u_2(1) = Q(\, y(1) - CBX_{D_1}u_2(0)\,)$*, with* $u_2(0)$ *obtained in i); else, for some* $y \in \operatorname{Im} D_2$ *such that* $d_H(Q(y(1) - CBX_{D_1}u_2(0)), y) =$ *rm* $\min d_H(Q(y(1) - CBX_{D_1}u_2(0)), \operatorname{Im} D_2)$*,* $u_2(1)$ *is a solution of* $D_2u_2(1) = y$*, with* $u_2(0)$ *obtained in i)*

$\vdots$

ℓ) Computing
$d_H(Q(\, y(\ell) - X_{D_1}(CA^{\ell-1}Bu_2(0) - CA^{\ell-2}Bu_2(1) - \ldots - CBu_2(\ell - 1)), \operatorname{Im} D_2\,))$*, with* $u_2(0)$*,* $u_2(1)$*, …,* $u_2(\ell-1)$ *obtained in i), ii), …,* $\ell - 1)$*.*

if
$d_H(Q(y(\ell) - X_{D_1}(CA^{\ell-1}Bu_2(0) - CA^{\ell-2}Bu_2(1) - \ldots - CBu_2(\ell - 1))), \operatorname{Im} D_2) = 0$*,* $u_2(\ell)$ *is a solution of:* $D_2u_2(l) = Q(y(\ell) - X_{D_1}(CA^{\ell-1}$ $Bu_2(0) - CA^{\ell-2}Bu_2(1) - \ldots - CBu_2(\ell-1)))$ *with* $u_2(0)$*,* $u_2(1)$*, …,* $u_2(\ell - 1)$ *obtained in i), ii), …,* $\ell - 1)$*;*
else, for some $y \in \operatorname{Im} D_2$ *such that*
$d_H(Q(y(\ell) - X_{D_1}(CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1))), y) =$ $\min d_H(Q(y(\ell) - X_{D_1}(CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell - 1))), \operatorname{Im} D_2)$*,* $u_2(l)$ *is a solution of:* $D_2.u_2(l) = y$*, with* $u_2(0)$*,* $u_2(1)$*, …,* $u_2(\ell - 1)$ *obtained in i), ii), …,* $\ell - 1)$*.*

4.4.3 Iterative decoding algorithm for parallel concatenation

When it comes to the decoding in a parallel concatenation case, we have to consider the issue under multiple angles. For this, we need to keep in mind that the input is the same for both codes, so the recovery of the input is only done once. And then later on, outputs coming from the two codes are summed up.

case 1 The first angle is to consider that in case of error, it occurred after the concatenation of outputs of both codes. The approach for this decoding method would be to follow the steps that are to detect the error, correct it and then dissociate each part coming from the different convolutional codes, in that order.

case 2 The second possibility is to consider that the mistake occurred in between each encoding and concatenation, but not after. Therefore, we would have to first of all try to dissociate both pieces, and then attempt to decode from every one of them.

case 3 The last option, that makes the decoding more complicated is if there are errors occurring somewhere between encoding and concatenation, and also after concatenation. In that case, there are multiple errors, and the decoding is more complex, and takes a longer time.

Example 4.4.7. For this case, we look at a parallel concatenated model.

In $\mathbb{F}_3$, we consider two convolutional codes $\mathcal{C}_1$ and $\mathcal{C}_2$, with respectively

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, D_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}, D_2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}.$$

After the parallel concatenation of those two codes, we get:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \end{pmatrix}, D = \begin{pmatrix} 2 & 2 \\ 0 & 0 \end{pmatrix}.$$

We will try to decode the sequence $y = (2\,1,\ 0\,2,\ 1\,1)$.

We consider the encoded sequence with the $\mathcal{C}_1$-code: $y_1 = (y_1(0), y_1(1), y_1(2))$, and the one with the $\mathcal{C}_2$-code $y_2 = (y_2(0), y_2(1), y_2(2))$.

In this particular case, we consider the case 1, where in case of an error, it occurred after concatenation.

Step 1: we set $x(0) = (0\,0)$

Step 2: we can observe that D does not have full rank; moreover, $\bar{y}(0) = (2\,1)$ does not belong to $\mathrm{Im}\,(D)$. $d_H((2\,1),(2\,0)) = 1$, which means that the error was in the second piece of the sequence; and $u(0)$ is either $(1\,0)$ or $(0\,1)$; we consider $u(0) = (1\,0)$.

Moving on to the dissociating part; having that: $D = D_2 + D_1$, then $Du(0) = (D_1 + D_2)u(0) = y_1(0) + y_2(0) = \left(\begin{pmatrix}1 & 2\\ 1 & 2\end{pmatrix} + \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\right)\begin{pmatrix}1\\ 0\end{pmatrix} = \begin{pmatrix}2\\ 0\end{pmatrix}$.

Which comes down to: $y_1(0) = (1\,1)$ and $y_2(0) = (1\,2)$.

Step 3: We have

$$\begin{pmatrix} CB & D\end{pmatrix}\begin{pmatrix}u(0)\\ u(1)\end{pmatrix} = \begin{pmatrix}y(1)\end{pmatrix}.$$

Then, we solve

$$\begin{pmatrix}2 & 2 & 2 & 2\\ 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\ 0\\ u(1)\end{pmatrix} = \begin{pmatrix}\bar{y}(1)\end{pmatrix} = \begin{pmatrix}0\\ 2\end{pmatrix}.$$

This comes down to solving $Du(1) = \begin{pmatrix}2 & 2\\ 0 & 0\end{pmatrix} u(1) = \begin{pmatrix}1\\ 2\end{pmatrix}$ $(1\,2)$ does not belong to $\mathrm{Im}(D)$; however, $(1\,0)$ does; and $d_H((1\,2),(1\,0)) = 1$, which means that the error is in the second piece of the sequence, and $u(1)$ is $(1\,1)$.

Moving on to the dissociating part; having that:
$y(1) = (C_1B_1 + C_2B_2)u(0) + (D_1 + D_2)u(1)$, then:
$y_1(1) = C_1B_1u(0) + D_1u(1)$ and $y_2(1) = C_2B_2u(0) + D_2u(1)$

Then we have: $y_1(1) = \begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} + \begin{pmatrix}1 & 2\\ 1 & 2\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix} = \begin{pmatrix}1\\ 1\end{pmatrix}$.

$y_2(1) = \begin{pmatrix}1 & 1\\ 2 & 2\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} + \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix} = \begin{pmatrix}2\\ 2\end{pmatrix}$

Which comes down to: $y_1(1) = (1\,1)$ and $y_2(1) = (2\,2)$.

Lastly, we have

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = y(2).$$

Then, we solve

$$\begin{pmatrix} 2 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ u(2) \end{pmatrix} = y(2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This comes down to solving $Du(2) = \begin{pmatrix} 2 & 2 \\ 0 & 0 \end{pmatrix}.u(2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $(1\,1)$ does not belong to $\mathrm{Im}(D)$. However, $d_H((1\,1),(1\,0)) = 1$, and the system is compatible with $u(2) = (1\,1)$. We decide to settle with $u(2) = (1\,1)$.

Moving on to the dissociating part; having that
$y(2) = (C_1A_1B_1 + C_2A_2B_2)u(0) + (C_1B_1 + C_2B_2)u(1) + (D_1 + D_2)u(2)$, then:
$y_1(2) = C_1A_1B_1u(0)+C_1B_1u(1)+D_1u(2)$ and $y_2(2) = C_2A_2B_2u(0)+C_2B_2u(0)+D_2u(1)$.

Then we have:

$$y_1(2) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$y_2(2) = \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

For this case, $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,2,\ 1\,1)$ there was errors during all transmission; there was 3 errors: $d_H = 3$. The original encoded sequence was: $y = (y(0), y(1), y(2)) = (2\,0\ 0\,0\ 1\,0)$.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (1\,0,\ 1\,1,\ 1\,1)$.

Example 4.4.8. For this case, we look at a parallel concatenated model.

In $\mathbb{F}_3$, we consider two convolutional codes $\mathcal{C}_1$ and $\mathcal{C}_2$, with respectively

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, D_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}, D_2 = \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix}.$$

After the parallel concatenation of those two codes, we get

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \end{pmatrix}, D = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}.$$

We try to decode the sequence using the second model of decoding: $\bar{y} = (2\,1\;0\,2\;1\,1)$.

In this particular case, we consider the case 2, where in case of an error, it occurred before concatenation.

Step 1: we set $x(0) = (0\,0)$

Step 2: We can observe that D does not have full rank; moreover, $y(0) = (2\,1)$ does not belong to $\operatorname{Im} D$. We clearly observe that an error occurred within the sequence. We decide to approach the supposed input sequence by computing the Moore-Penrose pseudoinverse if exists; otherwise a generalized inverse will do.

We have $\operatorname{rank} D = \operatorname{rank}(D^t) = \operatorname{rank}(DD^t) = \operatorname{rank}(D^tD) = 1$ Which means that D^+ exists; we finally get: $D^+ = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$

Then $u(0) = D^+y(0) = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ So, we get $u(0) = (0\,1)$.

For $u(0) = (0\,1)$, we get $y(0) = Du(0) = (D_2 + D_1)u(0) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$; moreover $(2\,0)$ belongs to $\operatorname{Im} D$. So, we consider that only one error occurred, in the second piece of the sequence before concatenation.

So we get

$$y_1(0) = D_1 u(0) = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

and

$$y_2(0) = D_2 u(0) = \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We can detect that there was an error either on $y_1(0) = (2\,2)$ or $y_2(0) = (0\,1)$.

Step 3: We have

$$\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} = \begin{pmatrix} y(1) \end{pmatrix}$$

Then, we solve

$$\begin{pmatrix} 2 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ u(1) \end{pmatrix} = \begin{pmatrix} y(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

This comes down to solving $Du(1) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} u(1) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$(1\,2)$ does not belong to $\operatorname{Im} D$; we clearly observe that an error occurred within the sequence. We already have $D^+ = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$;

Then $u(1) = D^+ \bar{y}(1) = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$. So, we get $u(1) = (0\,2)$.

For $u(1) = (0\,2)$, we get $Du(1) = (D_2 + D_1)u(1) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; moreover $(1\,0)$ belongs to $\operatorname{Im} D$. So, we will consider that only one error occurred, in the second piece of the sequence before concatenation.

So we get

$$y_1(1) = D_1 u(1) = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

and

$$y_2(1) = D_2 u(1) = \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

We can detect that there was an error either on $y_1(0) = (1\,1)$ or $y_2(0) = (0\,2)$.

Lastly, we have

$$\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} = \begin{pmatrix} y(2) \end{pmatrix}.$$

Then, we solve

$$\begin{pmatrix} 2 & 1 & 2 & 2 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 2 \\ u(2) \end{pmatrix} = \bar{y}(2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This comes down to solving $Du(2) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} u(2) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$.

Then, $u(2) = D^+ \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ So, we get $u(2) = (0\,1)$.

For $u(2) = (0\,1)$, we get $Du(2) = (D_2 + D_1)u(2) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$; moreover $(1\,0)$ belongs to $\mathrm{Im}(D)$. So, we consider that only one error occurred, in the second piece of the sequence before concatenation.

So we get

$$y_1(2) = D_1 u(2) = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

and

$$y_2(2) = D_2 u(1) = \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We can detect that there was an error either on $y_1(2) = (2\,2)$ or $y_2(2) = (0\,1)$.

For this case, $y = (y(0), y(1), y(2)) = (2\,1,\ 0\,2,\ 1\,1)$ there were errors during all transmission; there was 3 errors: $d_H = 3$. The original encoded sequence was $y = (y(0), y(1), y(2)) = (2\,0\ 1\,0\ 2\,0)$.

With $x(0) = 0$, the decoded sequence is $u = (u(0), u(1), u(2)) = (1\,0,\ 1\,1,\ 1\,1)$.

We consider that we are dealing with the first case.

Generalizing the following examples, we already know that the input is the same for both codes, and is given by the general decoding procedure.

Indeed, the procedure used for the general decoding procedure is the same as for the parallel concatenated case.

Proposition 4.4.13. *Let (A, B, C, D) be a representation of a parallel concatenated code of two convolutional codes: $\mathcal{C}_1(A_1, B_1, C_1, D_1)$, and $\mathcal{C}_2(A_2, B_2, C_2, D_2)$, over a field $\mathbb{F}_q$ with $q = q_1^{m_1}$, q_1 being prime. Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, the decoded sequence $(u(0), u(1), \ldots, u(\ell))$ input of the concatenated code $\mathcal{C}$ is obtained recursively as follows:*

a) Choosing $x(0) = 0$

b) i) Computing $d_H(y(0), \operatorname{Im} D)$;
if $d_H(y(0), \operatorname{Im} D) = 0$, $u(0)$ is a solution of: $D.u(0) = y(0)$;
else, for some $y \in \operatorname{Im} D$ such that $d_H(y(0), y) = \min d_H(y(0), \operatorname{Im} D)$, $u(0)$ is a solution of: $Du(0) = y$.

ii) Computing $d_H(y(1) - CBu(0), \operatorname{Im} D)$, with $u(0)$ obtained in i);
if $d_H(\, y(1) - CBu(0), \operatorname{Im} D) = 0$, $u(1)$ is a solution of:
$Du(1) = y(1) - CBu(0)$, with $u(0)$ obtained in i);
else, for some $y \in \operatorname{Im} D$ such that
$d_H(\, y(1) - CBu(0), y) = \min d_H(y(1) - CBu(0), \operatorname{Im} D)$, $u(1)$ is a solution of: $Du(1) = y$, with $u(0)$ obtained in i)

$\vdots$

ℓ) Computing
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D_1\,)$, with $u(0)$, $u(1)$, …, $u(\ell-1)$ obtained in i), ii), …, $\ell-1$).
if
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = 0$,
$u(\ell)$ is a solution of:

$D.u(l) = y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$
with $u(0)$, $u(1)$, ..., $u(\ell-1)$ *obtained in i), ii), ...,* $\ell-1$*);*
else, for some $y \in \operatorname{Im} D$ *such that*
$d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), y) = \min d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D)$,
$u(l)$ *is a solution of:* $Du(l) = y$, *with* $u(0)$, $u(1)$, ..., $u(\ell-1)$ *obtained in i), ii), ...,* $\ell-1$*).*

However, we can deduct the following result, in order to evaluate the output from each code:

Corollary 4.4.2. *Let* (A, B, C, D) *be a representation of a parallel concatenated code of two convolutional codes:* $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ *and* $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ *over a field* $\mathbb{F}_q$ *with* $q = q_1^m$, q_1 *being prime.*
Given the decoded input sequence $u(0), u(1), \ldots, u(\ell)$ *and choosing* $x(0) = 0$, *the output sequence* $y_1(0), y_1(1), \ldots, y_1(\ell)$ *(respectively* $y_2(0), y_2(1), \ldots, y_2(\ell)$*) of the code* $\mathcal{C}_1$ *(respectively* $\mathcal{C}_2$*) is obtained by:*
$y_1(j) = \sum_{i,j-1=0}^{\ell-1} C_1 A_1^{\ell-(\ell-(j-1))} B_1 u(i) + D_1 u(j)$
$(resp\ y_2(j) = \sum_{i,j-1=0}^{\ell-1} C_2 A_2^{\ell-(\ell-(j-1))} B_2 u(i) + D_2 u(j))$.

4.4.4 Iterative decoding algorithm for parallel with interleaver concatenation

When it comes to the decoding in a parallel concatenation case with interleaver, we have to consider the issue under multiple angles. For this, we need to keep in mind that the input is the same for both codes, so the recovery of the input is only done once. And then later on the outputs, the first coming form $\mathcal{C}_1$, and the second from $\mathcal{C}_2$ preceded by an interleaver, are summed up.

From the concatenation process used in parallel mode with interleaver, we can deduce the result, very similar to the parallel concatenated mode.

Proposition 4.4.14. *Let* (A, B, C, D) *be a representation of a parallel concatenated code of two convolutional codes* $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ *and* $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ *over a field* $\mathbb{F}_q$ *with* $q = q_1^m$, q_1 *being prime, with* $\mathcal{P}$ *the interleaver.*
Given the sequence $y = (y(0), y(1), \ldots, y(\ell))$, *the decoded sequence* $(u(0), u(1),$ ..., $u(\ell))$ *input of the parallel concatenated code with interleaver* $\mathcal{P}$ *is obtained recursively as follows:*

a) Choosing $x(0) = 0$

b) i) Computing $d_H(y(0), \operatorname{Im} D)$*;*
if $d_H(y(0), \operatorname{Im} D) = 0$, $u(0)$ *is a solution of:* $D.u(0) = y(0)$*;*
else, for some $y \in \operatorname{Im} D$ *such that* $d_H(y(0), y) = \min d_H(y(0), \operatorname{Im} D)$, $u(0)$ *is a solution of:* $Du(0) = y$.

ii) Computing $d_H(y(1) - CBu(0), \operatorname{Im} D)$, *with* $u(0)$ *obtained in i);*
if $d_H(\, y(1) - CBu(0), \operatorname{Im} D) = 0$, $u(1)$ *is a solution of:* $Du(1) = y(1) - CBu(0)$, *with* $u(0)$ *obtained in i);*
else, for some $y \in \operatorname{Im} D$ *such that* $d_H(\, y(1) - CBu(0), y) = \min d_H(y(1) - CBu(0), \operatorname{Im} D)$, $u(1)$ *is a solution of:* $Du(1) = y$, *with* $u(0)$ *obtained in i)*

$\vdots$

ℓ) Computing
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D_1\,)$, *with* $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$).
if
$d_H(\, y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D) = 0$,
$u(\ell)$ *is a solution of:*
$D.u(l) = y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1)$
with $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$*);*
else, for some $y \in \operatorname{Im} D$ *such that*
$d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), y) = \min d_H(y(\ell) - CA^{\ell-1}Bu(0) - CA^{\ell-2}Bu(1) - \ldots - CBu(\ell-1), \operatorname{Im} D)$,
$u(l)$ *is a solution of:* $Du(l) = y$, *with* $u(0)$, $u(1)$, …, $u(\ell-1)$ *obtained in i), ii), …,* $\ell - 1$).

However, we can deduct the following result, in order to evaluate the output from each code:

Corollary 4.4.3. *Let* (A, B, C, D) *be a representation of a parallel concatenated code with interleaver* $\mathcal{P}$ *of two convolutional codes:* $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ *and* $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ *over a field* $\mathbb{F}_q$ *with* $q = q_1^m$, q_1 *being prime.*
Given the decoded input sequence $u(0), u(1), \ldots, u(\ell)$ *and choosing* $x(0) = 0$, *the output sequence* $y_1(0), y_1(1), \ldots, y_1(\ell)$ *(respectively* $y_2(0), y_2(1), \ldots, y_2(\ell)$*) of the code* $\mathcal{C}_1$ *(respectively* $\mathcal{C}_2$*) is obtained by:*
$y_1(j) = \sum_{i,j-1=0}^{\ell-1} C_1 A_1^{\ell-(\ell-(j-1))} B_1 u(i) + D_1 u(j)$
$(resp\ y_2(j) = \sum_{i,j-1=0}^{\ell-1} C_2 A_2^{\ell-(\ell-(j-1))} B_2 \mathcal{P} u(i) + D_2 \mathcal{P} u(j))$

4.5 Output-observability matrix and Syndrome former matrix

In this section we are going to relate the output observability matrix with the syndrome former matrix used by Rosenthal and York [71], solving the decoding problem.

Let (A, B, C, D) be a realization of a convolutional code.

From the system

$$\begin{pmatrix} C & D & & & & \\ CA & CB & D & & & \\ CA^2 & CAB & CB & D & & \\ \vdots & & & \ddots & \ddots & \\ CA^{\ell} & CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} x(s) \\ u(s) \\ \vdots \\ u(s+\ell) \end{pmatrix} = \begin{pmatrix} y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix} \quad (4.8)$$

we can deduce the syndrome former matrix for the given code.

Proposition 4.5.1. *Suppose that $\ell \geq \delta$. By making elementary transformations to matrix equation* (4.8) *we can deduce the syndrome former matrix for the convolutional code.*

Proof. The system (4.8) can be rewritten as

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\ell} \end{pmatrix} x(s) + \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} u(s) \\ u(s+1) \\ \vdots \\ u(s+\ell) \end{pmatrix} = \begin{pmatrix} y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix}$$

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\ell} \end{pmatrix} x(s) = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} -u(s) \\ -u(s+1) \\ \vdots \\ -u(s+\ell) \end{pmatrix} + \begin{pmatrix} I & & \\ & \ddots & \\ & & I \end{pmatrix} \begin{pmatrix} y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix}$$

That can be written as

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^\ell \end{pmatrix} x(s) = \begin{pmatrix} D & & & & & I & & & \\ CB & D & & & & & I & & \\ CAB & CB & D & & & & & I & \\ \vdots & & & \ddots & \ddots & & & & \ddots \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D & & & & I \end{pmatrix} \begin{pmatrix} -u(s) \\ -u(s+1) \\ \vdots \\ -u(s+\ell) \\ y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix} \tag{4.9}$$

Now, and taking into account that $\ell \geq \delta$ there exist an invertible matrix $P \in Gl(p \times \ell, \mathbb{F})$, such that

$$P \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^\ell \end{pmatrix} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\delta-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \mathcal{O} \\ 0 \end{pmatrix}$$

where $\mathcal{O}$ is the observability matrix defined in (3.3).

Applying the matrix P to the matrix equation (4.9) we obtain

$$\left(\frac{\mathcal{O}}{0}\right)(x(s)) = \left(\frac{M_1 \quad , \quad M_2}{M_3 \quad , \quad M_4}\right) \left(\frac{\begin{matrix} -u(s) \\ \vdots \\ -u(s+\ell) \end{matrix}}{\begin{matrix} y(s) \\ \vdots \\ y(s+\ell) \end{matrix}}\right) \tag{4.10}$$

Then, $\begin{pmatrix} M_3 & , & M_4 \end{pmatrix}$ is the syndrome former matrix. □

Example 4.5.1. Considering the above example 4.2.2, the system (4.9) for this particular case is

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} (x(s)) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -u(0) \\ -u(1) \\ -u(2) \\ -u(3) \\ -u(4) \\ -u(5) \\ -u(6) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{pmatrix}$$

Now, taking

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The system is reduced to

$$\left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right) (x(s)) = \left(\begin{array}{ccccccc:ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c} -u(0) \\ -u(1) \\ -u(2) \\ -u(3) \\ -u(4) \\ -u(5) \\ -u(6) \\ \hline y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{array}\right)$$

So, the syndrome former matrix is

$$\left(\begin{array}{ccccccc,ccccccc} -1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

Chapter 5

Convolutional Codes and Steganography.

5.1 Introduction

In this chapter, we introduce an application of the convolutional coding theory in steganography. As we already know, block codes are largely used for dissimulation of information using the steganographic process, such as in [23]. Here, we suggest a steganographic protocol based on convolutional codes defined under the linear systems theory. As defined earlier, convolutional codes can be given by the quadruple (A, B, C, D) which works by considering an embedding protocol for a message m into a message u with as less distortion as possible.

5.2 Steganography

Steganography can be comparable to protection of communication, as it is known as a technique being used in order to protect some information to be exchanged by hiding its original existence, onto some digital files, could it be a photographies or videograms. As we know of cryptography as the technique and science behind the protection of messages and information to be transmitted, the idea for steganography is actually to prevent a nasty observer to even detect the need for that protection firstly, and it is also depending on the situations, as for instance in places where cryptography cannot be used. Sometimes, it is also possible to mix together both techniques for protection

of communication and information. The classic example known to illustrate use of a steganographic scheme is the prisoners problem exchanging messages under the surveillance of a warden.

5.2.1 Characteristics of a steganographic scheme

A steganographic scheme is characterized by some necessary conditions and components which are:

i) the choice of a communication support

ii) the message to be embedded

iii) embedding function

iv) extracting function

v) optional steganographic key-management

The embedding and extracting functions are as their names show consist of the functions responsible for hiding messages or information. For digital steganography, as it is in our case, the purpose is to hide or embed a sequence of bits in that digital cover, within some conditions such as making sure that the cover object does not show "perceptuable"distortion.
Knowing that within the digital world, the choice of covers is quite large (graphic files, messages, etc), but is then dictated by the nature of the information to embed. For that matter, the performance of a steganographic method can be assessed over a certain cover object mainly by its average distortion and its embedding rate.

For instance, a very popular method used in digital steganography is called Least Significant Bit (LSB) Steganography, and it consists into hiding information in a graphic file, by replacing the least significant bits of specifically selected pixels by message bits, in such a way that they are "visually imperceptible".

Definition 5.2.1. A digital steganographic scheme S of type $[k, n]$ over an alphabet A is a pair of functions:

$$\begin{aligned} emb: \quad & A^n \times A^k \longrightarrow A^n \\ rec: \quad & A^n \longrightarrow A^k \end{aligned}$$

such that

$$rec(emb(c, m)) = m \text{ for all } c \in A^n \text{ and } m \in A^k,$$

with m being the secret message, and c the cover vector.

The scheme is denoted as: $S(emb, rec)$.
We have: $c' = emb(c, m)$ and $rec(c') = m$

Then, the scheme S has the following parameters:

a) the cover length n

b) the embedding capacity k

c) the embedding radius ρ, defined by:

$$\rho = \max\{d(c, emb(c, m)) \mid c \in A^n, m \in A^k\}.$$

where d is the Hamming distance

d) the average number of embedding changes R_a, given by:

$$R_a = \frac{1}{q^{kn}} \sum d(c, emb(c, m))$$

where $q = \sharp A$

Proposition 5.2.1 ([58]). *Let $S = (emb, rec)$ be a steganographic scheme of type $[n, k]$ defined over the alphabet A. Then:*

1) the map rec is surjective;

2) for fixed $c \in A^n$, the map $emb(c, -) : A^k \longrightarrow A^n$ is injective.

In particular, $k \leq n$.

Proof. From the condition $rec(emb(c, m)) = m$. □

Knowing that the purpose of a stegoscheme is to embed as much information as possible, with as few changes as possible, we have the definition of a proper scheme;

Definition 5.2.2. A steganographic scheme $S = (emb, rec)$ is said to be proper if the number of changes produced in the cover is the minimum possible allowed by the recovering map.

$$d(c, emb(c, m)) = d(c, rec^{-1}(m)), \text{ for all } c \in A^n \text{ and } m \in A^k.$$

We have the following proposition:

Proposition 5.2.2 ([58])**.** *Let $S = (emb, rec)$ be a steganographic scheme of type $[n, k]$ over A. There exists a proper stegoscheme $S^* = (emb^*, rec)$ of the same type $[n, k]$ such that $R_a(S^*) \leq R_a(S)$.*

5.3 Steganography and Coding

There are some interesting steganographic protocols that have already been defined from coding theory. Considering the fact that error-correcting codes are used in order to detect and/or correct errors, during data transfer. If we consider for instance some of the methods involving the existence of the parity check matrix, we can implement the syndrome coding. Considering a steganographic protocol within the spatial domain of gray scale image, inspired by [57]. This approach suggests to divide the cover block into blocks of equal sizes.

For instance, let us consider the following protocol having the cover object v whose LSB values are given by $v = v_0, v_1, \ldots, v_n$ over $\mathbb{F}_2^n$, the message m to embed $m = m_0, m_1, \ldots, m_t$ with $t < n$ over $\mathbb{F}_2^t$, the code given by its parity matrix H.
Embedding m into v produces the stego object $r = r_0, r_1, \ldots, r_n$, given by the relation:

$$m = r \times H^t \tag{5.1}$$

In order to extract m, it is the same equation 5.1 that is used. After embedding, some of the bits of the cover block are modified (either 0 or 1); if we consider e, the flip pattern representing the modified bits of that cover block, by having every object of the protocol given by its polynomial representation, the stego object is given by:

$$r(X) = v(X) + e(X) \tag{5.2}$$

From both equations 5.1 and 5.2, we have:

$$m - v \times H^t = e \times H^t \tag{5.3}$$

which gives us the extraction formula.

Example 5.3.1. In the $F5$ algorithm [88], the technique used with an $[n, n-k, 1]$-code consists of embedding k bits into an n-length cover sequence by changing at most 1 bit.

This method is called the syndrome coding, and from a steganographic point of view, we need to find a minimal number of flips of $e(X)$ to decrease distortion.

5.4 Steganography and convolutional coding

Before going any further, let us recall some notions involved within the construction of the steganographic protocol based on coding.

5.4.1 The purposes and interest

From what we already get out of the traditional steganographic procedure, the idea is to suggest an efficient steganographic protocol that is implementable on convolutional encoding/decoding. As we know that there exists several steganographic protocols defined over error-correcting block codes, within the decoding method actually used to detect and correct errors, in order to introduce a minimum amount of errors, as few as possible [58]. We are using the same approach on our own steganographic model here, with the twist and particularity of convolutional codes, which requires for the sequential characteristic of the implementation, for instance embedding of a sequence while transmission of a message, file or images during an undetermined, or semi-infinite sequence of time. Indeed, the general plan is to introduce "as few distortion as possible", onto the cover sequence, in order to embed another digital sequence (preferably, of less length).

As we are trying to get there, some key points we have to cover are:

1. the conditions for the steganographic scheme to be established, which means *rec* and *emb* functions to be right, and well described.
2. the conditions for the modified subsections, which is the bound on the flipping bits, the bits that are being altered while embedding, in order to alter the least bits possible;

3. the classical bound of imperceptibility, as far as the embedding radius. Indeed, we have to give conditions on the convolutional code in order to keep the imperceptibility of the change on the cover, independently of the cover or the message to hide;

4. the interest and benefits, as compared to the steganography, based on block codes;

As for the case for some steganographic schemes based on codes in block, the embedding and recovering functions are based on the decoding procedure; we will be working around the same idea in order to implement our own functions in this case.

5.4.2 Proposition of a Stegosystem based on convolutional codes

In this subsection, we show our construction and implementation of the steganographic scheme based on convolutional codes, most specifically based on the convolutional codes approach based on linear systems. Before going any further, let us introduce some notions we use throughout the process;

Let (A, B, C, D) be a convolutional code. We denote by τ the minimum number of linearly-dependent columns of D.

By analogy to the block coding theory, τ can be related to the minimal distance of a block code whose control matrix is represented by D.

As already defined earlier, we consider our convolutional codes, by their realization representation given by the quadruple of matrices (A, B, C, D). For this scheme, we are using the decoding method and protocol implemented for this specific representation of convolutionals, by involving the output-observability

matrix T_ℓ given by the matrix:

$$T_\ell = \begin{pmatrix} C & D & & & & \\ CA & CB & D & & & \\ CA^2 & CAB & CB & D & & \\ \vdots & & & \ddots & \ddots & \\ CA^\ell & CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}. \tag{5.4}$$

Indeed, decoding a system for this type of system consists of solving the system:

$$T_\ell \begin{pmatrix} x(0) \\ u \end{pmatrix} = y. \tag{5.5}$$

We recall that it is usual to consider the initial state of the system $x(0) = 0$, as in our case for instance; therefore, our new output-observability matrix is reduced to:

$$\widehat{T}_{\ell-1} = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \tag{5.6}$$

In order to do the embedding, the process goes by considering the output-observability matrix as the control matrix. The model of steganography we build is inspired by the (A, B, C, D)-representation of the convolutional code. Knowing that we are normally using its structure on convolutional codes for the decoding, step by step, when it comes down to steganography it is our base for the embedding function. In this case, we decide to approach it in a sequential fashion which means that for each step of the time-related steganographic process, at each $t = 1, \dots, \ell$, the protocol consists of "embedding"the message sequence, by altering lightly the cover sequence with some error, in order to build the stego-sequence. In order to do so, we need to figure out the best sequence corresponding to the flipping bits that minimizes the modification, which corresponds to the coset leader of the list of potential "error vectors $e(t)$" for embedding $m(t)$ in $u(t)$ by the formula: $u(t) + e(t)$.

When it comes to the retrieval of the hidden message, that's when the actual output-observability matrix explicitly appears for the solving of the corresponding equation. Indeed, the recovery process of the embedded message consists of the encoding of the stego-sequence. Which is an analogy of

the syndrome steganographic protocol method, in order to retrieve the hidden message, we extract at each step from the stego-sequence, using the control "block of matrices", each part of the embedded message.

Definition 5.4.1. The quasi-syndrome denoted by s is the value from which we choose the estimated pertubation e for embedding m into u. It is given by: $s = He$.

It can be related to the syndrome given within the stego-codes, based on block coding.

The following algorithm provides the method for the embedding process.

Embedding algorithm for embedding function emb
Input: Message m, Cover sequence u
Output: Stego-sequence St

if rank D is row maximal **then**
  for t:=0 to ℓ **do**
    $s(t) = m(t) - Du(t) - \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k))$;
    list $sysdTableD = \{e(t)\}$;
    compute $\min_{e \in \text{list}} w(e(t))$;
    pick and store one corresponding $e(t)$;
    $St(t) = u(t) + e(t)$;
  end for
  $St = (St(0), \ldots, St(\ell))$
else
  Choose an adequate code with D corresponding
end if

The next algorithm is used to extract the embedded message.

Extracting algorithm for recovery function rec

$m = \widehat{T}_{\ell-1}\left(u + e\right)$
for t:=0 to ℓ **do**
  $m(t) = Du(t) + De(t) + \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k))$;
end for
$m(t) = (m(0), \ldots, m(\ell))$

Example 5.4.1. In $\mathbb{F}_2$, let us consider the code $\mathcal{C}(A, B, C, D)$ defined by :

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

We observe that the system is output-observable.

Let us consider the message to be embedded be $m = (m(0), m(1), m(2))$, and is done so within those sections, at each step, in a cover sequence denoted by u.

Let us consider $m = (10,\ 00,\ 01)$

We consider that knowing each input sequence u, we will try to find out characteristics of each sequence e of the flipping bits that were added to u, in order to embed the message m. The corresponding quasi-syndrome is denoted by s.

Consider the decoding matrix for the convolutional codes given by:

$$\widehat{T}_{\ell-1} = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}$$

At each step here: $0, 1, \dots, \ell$, we will be try to evaluate our error value e

We already have: $m = \widehat{T}_{\ell-1}\left(u + e\right)$

As inputs, we have m and u.

First of all, let us assess the coset of potential errors e; as a general formula, m and u are given by: $m(t) = \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k)) + D(u(t) + e(t))$; which means that:

$$De(t) = s(t) = m(t) - \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k)) - Du(t)$$

Going through all possible cases of syndromes (or cover sequences), we get the coset of errors e.

s	e
0 0	0 0 0
	1 0 0
1 0	0 1 1
	1 1 1
0 1	0 0 1
	1 0 1
1 1	0 1 0
	1 1 0

Figure 5.1: Syndrome Table 5.4.1

Therefore, at each step, there is always a sequence e that can be used for embedding such that: $w(e) \leq 2$.

Let us work with $u = (111\ 010\ 001)$ having $m = (10,\ 00,\ 01)$.

At step $t = 0$, we have: $D(u(0) + e(0)) = m(0)$
Then, $s = (0\,0)$, and $e(0) = (0\,0\,0)$

At step $t = 1$, we have: $D(u(1) + e(1)) = m(1) - CB(u(0) + e(0))$
Then, $s = (0\,0)$, we pick $e(0) = (0\,0\,0)$ and $e(1) = (0\,0\,0)$

At step $t = 2$, we have: $D(u(2) + e(2)) = m(2) - CAB(u(0) + e(0)) - CB(u(1) + e(1))$
Then, $s = (1\,0)$, and $e(2) = (0\,1\,1)$.

Then, for $u = (1\,1\,1,\ 0\,1\,0,\ 0\,0\,1)$, we can embed $m = (10,\ 00,\ 01)$ with the flip pattern: $e = (0\,0\,0,\ 0\,0\,0,\ 0\,1\,1)$.

For this operation, we embedded 6 bits in a 9-length cover sequence by changing 2 bits.

Functions embedding and recovery

In order to define our convolutional code for the proper stegoscheme, according to our protocol, the following necessary condition is requested.

Proposition 5.4.1. *Let (A, B, C, D) be a representation of a convolutional code $\mathcal{C}$, with $A \in M_\delta(\mathbb{F})$, $B \in M_{\delta\times k}(\mathbb{F})$, $C \in M_{p\times\delta}(\mathbb{F})$, $D \in M_{p\times k}(\mathbb{F})$ (with $D \neq 0$ and $p = n - k$). Let $p < k$.*
A necessary condition for building a stegoscheme from $\mathcal{C}$ is that $\operatorname{rank} D$ *be row maximal.*

Conditions on the modified subsections of the cover

Here are some conditions applying to the modification of the cover sequence when embedding.

Proposition 5.4.2. *Let (A, B, C, D) be a representation of a convolutional code $\mathcal{C}$ for a steganographic scheme S. Then, at each step t of the convolutional sequence, the flipping sequence e introduced for embedding m in u can be given by the formula:*

$$De(t) = m(t) - Du(t) - \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k))$$

Proof. Having that the embedding of m is given by: $m = \widehat{T}_{\ell-1}\left(u + e\right)$, and having: $\widehat{T}_{\ell-1} = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}$, we can deduce the result. □

Classical bound of imperceptibility

Proposition 5.4.3. *Let (A, B, C, D) be a representation of a convolutional code $\mathcal{C}$ for a steganographic scheme S. Let us consider embedding m in u, with error sequence e. Let D have all of its columns non-zero and distinct.*

Then, for each step t of the convolutional sequence: $\exists\, e(t)$ such that $w(e(t)) \leq d-1$.

Proof. Considering $\widehat{T}_{\ell-1}$ the control matrix, the embedding of m is given by: $m = \widehat{T}_{\ell-1}\big(u+e\big)$;

at each step t, we have: $De(t) = s(t)$.

Let us consider D_j the columns of D; knowing that d is the minimal number of linearly-dependent columns of D, for all t, for $s(t) \neq 0$, there exists n columns D_j such that: $\sum_{j=1}^{n} D_j = s(t) = De(t) \neq 0$, with $n \leq \tau - 1$; therefore, $w(e(t)) = n \leq \tau - 1$; and for $s(t) = 0$, there is always $e(t) = 0$ which verifies: $w(e(t)) = 0 \leq \tau - 1$.

From there, we can deduce the result. □

This proposition follows from the precedent one.

Proposition 5.4.4. *Let us consider a steganographic scheme S given by a convolutional code (A, B, C, D), and functions emb and rec. Let D have all of its columns non-zero and distinct. Then, within an ℓk-length cover sequence, we can embed at most ℓp-length message by modifying at most $\ell(\tau - 1)$ bits*

Example 5.4.2. In $\mathbb{F}_2$, let us consider the code $\mathcal{C}(A, B, C, D)$ defined by :

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, D = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

We observe that the system is output-observable.

Let us consider the message to be embedded be $m = (m(0), m(1), m(2))$, and is done so within those sections, at each step, in a cover sequence denoted by u.

Let us consider $m = (111,\ 100,\ 001)$

We consider that knowing each input sequence u, we will try to find out characteristics of each sequence e of the flipping bits that were added to u, in order to embed the message m. The corresponding quasi-syndrome is denoted by s.

Consider the decoding matrix for the convolutional codes given by:

$$\widehat{T}_{\ell-1} = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}$$

At each step here: $0, 1, \dots, \ell$, we will be try to evaluate our error value e

We already have: $m = \widehat{T}_{\ell-1}\left(u + e\right)$

As inputs, we have m and u.

First of all, let us assess the coset of potential errors e; as a general formula, m and u are given by: $m(t) = \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k)) + D(u(t) + e(t))$; which means that:

$$De(t) = s(t) = m(t) - \sum_{k=0}^{t-1} CA^{t-1-k}B(u(k) + e(k)) - Du(t)$$

Going through all possible cases of syndromes (or cover sequences), we get the coset of errors e.

Therefore, at each step, there is always a sequence e that can be used for embedding such that: $w(e) \leq d - 1$ $(d = 3)$.

Let us work with $u = (11011\ 01000\ 10101)$ having $m = (111,\ 100,\ 001)$.

At step $t = 0$, we have: $D(u(0) + e(0)) = m(0)$
Then, $s = (1\,0\,1)$, and $e(0) = (1\,0\,0\,0\,0)$

At step $t = 1$, we have: $D(u(1) + e(1)) = m(1) - CB(u(0) + e(0))$
Then, $s = (0\,0\,0)$, we pick $e(0) = (1\,0\,0\,0\,0)$ and $e(1) = (0\,0\,0\,0\,0)$

At step $t = 2$, we have: $D(u(2) + e(2)) = m(2) - CAB(u(0) + e(0)) -$

s	e (w < d)	e (w ≥ d)
000	00000	00111
		11110
		11001
001	00001	11111
	11000	
	00110	
010	00010	11100
	00101	11011
100	01000	01111
	10001	
011	00011	11010
	00100	11101
101	01001	01110
	10000	10111
110	01010	01101
	10100	10011
111	01100	01011
	10010	10101

Figure 5.2: Syndrome Table 5.4.2

$CB(u(1) + e(1))$
Then, $s = (0\,0\,0)$, and $e(2) = (0\,0\,0\,0\,0)$.

Then, for

$$u = (1\,1\,0\,1\,1,\ 0\,1\,0\,0\,0,\ 1\,0\,1\,0\,1),$$

we can embed

$$m = (111,\ 100,\ 001)$$

with the flip pattern:

$$e = (1\,0\,0\,0\,0,\ 0\,0\,0\,0\,0,\ 0\,0\,0\,0\,0).$$

For this operation, we embedded 9 bits in a 15-length cover sequence by changing only 1 bits.

Interest and benefits compared to steganography based on block codes

Example 5.4.3. Let us consider a steganographic protocol based on a block code given by its transfer matrix:

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Let us consider a set of messages m to be embedded in a set of cover sequences u given by: $m(0) = (111)$ in $u(0) = (11011)$, $m(1) = (100)$ in $u(1) = (01000)$, and $m(2) = (001)$ in $u(2) = 10101$ one after the other, with this steganographic process.

Given the syndrome table:

The flipping sequence at each step is given by: e.

Embedding m is given by:

$$m(X) = H(u(X) + e(X)).$$

Then, $m(0) = H(u(0)+e(0))$; which means that: $He(0) = m(0)-Hu(0) = s(0)$; $s(0) = (1\,0\,1)$, and we pick $e(0) = (1\,0\,0\,0\,0)$

s	e (w < d)	e (w ≥ d)
000	00000	00111
		11110
		11001
001	00001	11111
	11000	
	00110	
010	00010	11100
	00101	11011
100	01000	01111
	10001	
011	00011	11010
	00100	11101
101	01001	01110
	10000	10111
110	01010	01101
	10100	10011
111	01100	01011
	10010	10101

Figure 5.3: Syndrome Table 5.4.3

s	e
0 0	0 0 0
	1 0 0
1 0	0 1 1
	1 1 1
0 1	0 0 1
	1 0 1
1 1	0 1 0
	1 1 0

Figure 5.4: Syndrome Table 5.4.4

As well as for $e(1)$, we have $De(1) = m(1) - Hu(1) = s(1)$; $s(1) = (0\,0\,0)$, and we have $e(1) = (0\,0\,0\,0\,0)$

As well as for $e(2)$, we have $De(2) = m(2) - Hu(2) = s(2)$; $s(2) = (1\,1\,0)$, and we pick $e(2) = (0\,1\,0\,1\,0)$

For the whole sequences put together, we have:
For $u = (1\,1\,0\,1\,1,\ 0\,1\,0\,0\,0,\ 1\,0\,1\,0\,1)$, we can embed $m = (111,\ 100,\ 001)$ with the flip pattern: $e = (1\,0\,0\,0\,0,\ 0\,0\,0\,0\,0,\ 0\,1\,0\,1\,0)$.

Working with the convolutional code, as previously, we have: for the same $u = (1\,1\,0\,1\,1,\ 0\,1\,0\,0\,0,\ 1\,0\,1\,0\,1)$, we can embed $m = (111,\ 100,\ 001)$ with the flip pattern: $e = (1\,0\,0\,0\,0,\ 0\,0\,0\,0\,0,\ 0\,0\,0\,0\,0)$.

When compared to the convolutional case, we are flipping 2 more bits.

Example 5.4.4. Let us consider a steganographic protocol based on a block code given by its transfer matrix:

$$H = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Let us consider a set of messages m to be embedded in a set of cover sequences u given by: $m(0) = (10)$ in $u(0) = (111)$, $m(1) = (00)$ in $u(1) = (010)$, and $m(2) = (01)$ in $u(2) = (001)$ one after the other, with this steganographic process.

Given the syndrome table:

The flipping sequence at each step is given by: e.

Embedding m is given by: $m(X) = H(u(X) + e(X))$.

Then, $m(0) = H(u(0)+e(0))$; which means that: $He(0) = m(0)-Hu(0) = s(0)$; $s(0) = (0\,0)$, and we pick $e(0) = (0\,0\,0)$

As well as for $e(1)$, we have $De(1) = m(1) - Hu(1) = s(1)$; $s(1) = (1\,1)$, and we have $e(1) = (0\,1\,0)$

As well as for $e(2)$, we have $De(2) = m(2) - Hu(2) = s(2)$; $s(2) = (0\,0)$, and we pick $e(2) = (0\,0\,0)$

For the whole sequences put together, we have:
for $u = (1\,1\,1,\ 0\,1\,0,\ 0\,0\,1)$, we can embed $m = (10,\ 00,\ 01)$ with the flip pattern: $e = (0\,0\,0,\ 0\,1\,0,\ 0\,0\,0)$

Working with the convolutional code, as previously, we have:
for the same $u = (1\,1\,1,\ 0\,1\,0,\ 0\,0\,1)$, we can embed $m = (10,\ 00,\ 01)$ with the flip pattern: $e = (0\,0\,0,\ 0\,0\,0,\ 0\,1\,1)$

When compared to the convolutional case, we are flipping 1 less bit.

Example 5.4.5. Let us consider a steganographic protocol based on a block code given by its transfer matrix:

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Let us consider a set of messages m to be embedded in a set of cover sequences u given by: $m(0) = (110)$ in $u(0) = (1100110)$, $m(1) = (010)$ in $u(1) = (0010011)$, and $m(2) = (011)$ in $u(2) = (1001010)$ one after the other, with this steganographic process.

Given the syndrome table:

The flipping sequence at each step is given by: e.

Embedding m is given by: $m(X) = H(u(X) + e(X))$.

Then, $m(0) = H(u(0)+e(0))$; which means that: $He(0) = m(0)-Hu(0) =$

s	e (w < d)
000	0000000
001	0010000
010	0100000
100	1000000
011	0000100
	0110000
101	0000001
	1010000
110	0001000
	1100000
111	0000010
	0011000

Figure 5.5: Syndrome Table 5.4.5

$s(0)$; $s(0) = (1\,0\,0)$, and we pick $e(0) = (1\,0\,0\,0\,0\,0\,0)$

As well as for $e(1)$, we have $De(1) = m(1) - Hu(1) = s(1)$; $s(1) = (0\,0\,1)$, and we have $e(1) = (0\,0\,1\,0\,0\,0\,0)$

As well as for $e(2)$, we have $De(2) = m(2) - Hu(2) = s(2)$; $s(2) = (1\,1\,0)$, and we pick $e(2) = (0\,0\,0\,1\,0\,0\,0)$

For the whole sequences put together, we have:
for $u = (1\,1\,0\,0\,1\,1\,0,\ 0\,0\,1\,0\,0\,1\,1,\ 1\,0\,0\,1\,0\,1\,0)$, we can embed $m = (110,\ 010,\ 011)$ with the flip pattern: $e = (1\,0\,0\,0\,0\,0\,0,\ 0\,0\,0\,1\,0\,0\,0\,0,\ 0\,0\,0\,1\,0\,0\,0)$

Let us consider instead the convolutional code, given by:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

At step $t = 0$, we have: $D(u(0) + e(0)) = m(0)$
$s(0) = (1\,0\,0)$, and we pick $e(0) = (1\,0\,0\,0\,0\,0\,0)$

At step $t = 1$, we have: $D(u(1) + e(1)) = m(1) - CB(u(0) + e(0))$
Then, $s = (0\,1\,1)$, we pick $e(0) = (1\,0\,0\,0\,0\,0\,0)$ and $e(1) = (0\,0\,0\,0\,1\,0\,0)$

At step $t = 2$, we have: $D(u(2) + e(2)) = m(2) - CAB(u(0) + e(0)) - CB(u(1) + e(1))$
Then, $s = (0\,1\,1)$, and $e(2) = (0\,0\,0\,0\,1\,0\,0)$.

for the same $u = (1\,1\,0\,0\,1\,1\,0,\ 0\,0\,1\,0\,0\,1\,1,\ 1\,0\,0\,1\,0\,1\,0)$, we can embed $m = (110,\ 010,\ 011)$ with the flip pattern: $e = (1\,0\,0\,0\,0\,0\,0,\ 0\,0\,0\,0\,1\,0\,0,\ 0\,0\,0\,0\,1\,0\,0)$

When compared to the convolutional case, we are flipping 3 bits, just as for the block code case.

If we try to assess in terms of relationship between all error vectors, let's look at what we have at each step:

At Step $t = 0$, $De(0) = m(0) - Du(0) < \tau$;

At step $t = 1$, $\begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} e(0) \\ e(1) \end{pmatrix} = m(1) - \begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix}$; which means that:

$$\begin{pmatrix} e(0) \\ e(1) \end{pmatrix} = X \begin{pmatrix} CB & D \end{pmatrix} \left(m(1) - \begin{pmatrix} CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} \right)$$

which means that: $w(e(0)) + w(e(1)) = C_1$, C_1 being a constant;

At step $t = 2$, $\begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} e(0) \\ e(1) \\ e(2) \end{pmatrix} = m(2) - \begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix}$;

which means that:

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \end{pmatrix} = X_{\begin{pmatrix} CAB & CB & D \end{pmatrix}} \left(m(2) - \begin{pmatrix} CAB & CB & D \end{pmatrix} \begin{pmatrix} u(0) \\ u(1) \\ u(2) \end{pmatrix} \right)$$

which means that: $w(e(0)) + w(e(1)) + w(e(2)) = C_2$, C_2 being a constant;

We can relate, using the same process, weights of all flipping sequences from $t = 0$ to $t = \ell$ from an analogical analysis.

Chapter 6

Conclusion and future work

In this thesis, we studied the plurality of convolutional codes. In fact, diverse definitions for convolutional codes can be found in the literature as suggested in a survey realized by Rosenthal in [66], and among all of the possible definitions, we picked the most suitable one for connection with the linear systems theory. Our ultimate purpose here is to comprehend all aspects of the convolutional codes, under the material and tools of the linear systems theory, as far as it concerns the algebraic aspect, coding or decoding, as well as the control theory properties involved.

First of all, we worked on the encoding aspect of the convolutional codes; doing so, we established a new realization building algorithm, to concretely express convolutional codes in terms of linear systems theory, and also constructed concatenated codes, following different models already existing. This algorithm introduced a less complicated, more intuitive approach to computation of our quadruple (A, B, C, D) in order to put in perspective the input-state-output representation from any convolutional code polynomial encoding matrix.

Within the possible construction of convolutional codes, we also explored the concatenation, which brings various benefits. We also worked on the control properties of those convolutional codes; we also showed, enhanced conditions to meet the control properties for any specific construction of those convolutional codes, such as controllability, observability and output-observability. Such as in other work on control properties, we remarked that it isn't trivial to extract those interesting conditions to match the control properties; however we managed to do so.

We worked on the decoding problem, by suggesting new methods and algorithms to solve the decoding of convolutional codes, based on terms of the linear systems. In fact, knowing that the Viterbi algorithm, is one of those really well known and used, decoding algorithms; with the construction based on algebraic terms, coming from the input-output representation, we were able to suggest easier methods to solve that complex problem of decoding, especially within the context of time-related communication. We noticed a tremendous saving of time, and more manageable computation process. With our methods also, we can directly get to the input, and we suggest the option to either detect the error(s), or correct, or both. Even though they are most-likelihood methods as well, the complexity for decoding is way more interesting, within any Galois Field, when compared to the algorithm used in Viterbi's for instance.

We also developed some new steganographic models, based on the representation of convolutional codes within the linear systems theory. In fact, the idea resides on considering the output-observability matrix, along with the encoding/decoding procedures used for the convolutionals. Indeed, the embedding and recovery maps inspired by this algebraic computational method, enable us to implement steganography for time-related transactions, for instance for protection of communication within transactions during an unspecified time, which is relatively new. That method revealed interesting results, such as an ability to hide a tremendous amount of information, with very little distortion, with specific conditions, which shows to abound in possibilities for that matter.

Bibliography

[1] B.M. Allen, *Linear Systems Analysis and Decoding of Convolutional Codes*. Ph.D. thesis, Department of Mathematics, University of Notre Dame, Indiana, USA (June 1999).

[2] W. Boumerdassi, E. Collange & Team Space Busters, *Turbo codes Encoding/Decoding & EXIT charts*, Georgia Tech Atlanta, 2010.

[3] F. Bavaud, J.-C. Chappelier, J. Kohlas, *An Introduction to Information Theory and Applications*. UniFr course, version 2.04, pp 9-11, (2005).

[4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, *Parallel concatenated trellis coded modulation*, in Proc. IEEE Int. Conf. Communications, vol. 2, Dallas, TX, pp. 974-978, June 1996.

[5] C. Berrou, A. Glavieux and P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding: turbocodes*, ICC 1993, Geneva, Switzerland, pp. 1064-1070, May 1993.

[6] G. Battail, C. Berrou and A. Glavieux, *Pseudorandom recursive convolutional coding for near capacity performance*, GLOBECOM 1993, Houston, Texas, USA, pp. 23-27.

[7] S.A. Barbulescu, *Iterative Decoding Of Turbo Codes and Other Concatenated Codes, A dissertation*, School of Electronic Engineering; University of South Australia, 1996.

[8] Bassoli et al.: *Network Coding Theory: A Survey*, IEEE Communications Surveys & Tutorials, Accepted for Publication, (2010).

[9] E. Berlekamp, editor. *Key Papers in the Development of Coding Theory*. IEEE Press, New York, 1974.

[10] E. Berlekamp. *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[11] R.E. Blahut. *Theory and Practice of Error Control Codes*, Addison Wesley, Reading, Mass., 1987.

[12] J-J. Climent, V. Herranz, C. Perea, *A first approximation of concatenated convolutional codes from linear systems theory viewpoint*, Linear Algebra and its Applications **425**, pp. 673-699, (2007).

[13] J-J. Climent, V. Herranz, C. Perea, *Linear system modelization of concatenated block and convolutional codes*, Linear Algebra and its Applications, **429**, pp. 1191-1212, (2008).

[14] C.-T. Chen, *Linear System Theory and Design.* Oxford, U.K.: Oxford Univ. Press, (1999).

[15] D. Divsalar and R.J. McEliece, *On the design of generalized concatenated coding systems with interleavers*, TMO PR 42-134, (1998).

[16] Y. Denneulin, J.-L. Roch, E. Tannier, *Théorie des codes*, pp. 41-46, (January 2000).

[17] D. Divsalar and F. Pollara, *On the design of turbo codes*, TMO PR 42-123, (1995).

[18] J-G. Dumas, J-L. Roch, E. Tannier, S. Varrette, *Théorie des codes: compression, cryptage, correction*, Dunod, pp. 13-17, (2007).

[19] P. Elias, *Coding for noisy channels*, IRE Conv. Rec. 4 , pp. 37-46, (1955).

[20] G.D. Forney, *Convolutional codes*, Algebraic structure, IEEE Trans. Information Theory (1970).

[21] G.D. Forney, Jr. *On decoding BCH codes*, IEEE Trans. Information Theory, vol. IT-11, pp. 549-557, (October 1965).

[22] Ch. Fragouli, R.D. Wesel, *Convolutional Codes and Matrix Control Theory*, Proceedings of the 7th International Conference on Advances in Communications and Control, Athens, Greece, (1999).

[23] J. Fridrich, *Steganography in Digital Media - Principles, Algorithms, and Applications*, Cambridge Univ. Press, 2009.

[24] R.G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, (1968).

[25] Mª I. García-Planas, M.D. Magret, *An alternative System of Structural Invariants for Quadruples of Matrices*, Linear Algebra and its Applications **291**, (1-3), pp. 83-102, (1999).

[26] M.I. Garcia-Planas, D. Magret, M.E. Montoro, *Two parametric quasi-cyclic codes as hyperinvariant subspaces*. Cybernetics and physics Journal, **2**, (2), pp. 90-96, (2013).

[27] Mª I. García-Planas, El M. Souidi, L.E. Um, *Convolutional codes under linear systems point of view. Analysis of output-controllability*. WSEAS Transactions on Mathematics. vol. 11, (4), pp. 324-333, (2012).

[28] Mª I. Garcia-Planas, El M. Souidi, L.E. Um. *Convolutional codes under control theory point of view. Analysis of output-observability*. Recent Advances in Circuits, Communications & Signal Processing, pp. 131-137, (2013).

[29] M. I. Garcia-Planas, El M. Souidi, L. E. Um. *Decoding Algorithm for Convolutional Codes under Linear Systems Point of View*. Recent Advances in Circuits, Systems, Signal Processing and Communications, (2014), pp. 17-24.

[30] Mª I. García-Planas, S. Tarragona, *Output observability of time-invariant singular linear systems*, PHYSCON 2011, Léon, Spain, (2011).

[31] H. Gluesing-Luerssen and F.-L. Tsang, *A matrix ring description for cyclic convolutional codes*, Adv. Math. Commun., vol. 2, no. 1, pp. 55-81, (2008).

[32] V. Guruswami, *List Decoding of Error-Correcting Codes*, Department of Electrical Engineering and Computer Science, M.I.T , (September 2001).

[33] R. Hamming, *Coding and Information Theory*, Prentice Hall, (1980).

[34] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge Univ. Press, 2003.

[35] R. Hutchinson, J. Rosenthal, R. Smarandache, *Convolutional codes with maximum distance profile*, Systems Control Lett. 54 (1), pp. 53-63, (2005).

[36] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge Univ. Press, 2003.

[37] R. V. L. Hartley. *Transmission of information*. Bell System Tech. J.,7 pp 535-538 (1928).

[38] M. Hautus, *Controllability and observability condition for linear autonomous systems*, Proceedings of Nedderlandse Akademie voor Wetenschappen, Series A 72, pp. 443-448, (1969).

[39] A. Hocquenghem, *Codes correcteurs dérreurs*, Chiffres(Paris)2, pp. 147-156 (1959).

[40] W.C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge Univ. Press, (2003).

[41] R. Johannesson and K.Sh. Zigangirov, *Fundamentals of Convolutional Coding*. New York: IEEE Press, (1999).

[42] G.A. Jones, J.M. Jones, *Information and Coding Theory*, Springer-Verlag London 2000, pp. 97-113, (2000).

[43] H. Jouhari, *New Steganographic Schemes using Binary and Quaternary Codes*, Ph.D thesis, Université Mohammed V-Agdal, Morocco; (2013)

[44] H. Jouhari and El M. Souidi, *Application of Cyclic Codes over $\mathbb{Z}_4$ in Steganography*. Journal of Applied Mathematical Sciences, vol.6, N139, pp 6911-6925, (2012).

[45] H. Jouhari and El M. Souidi, *Steganographic Scheme Using The Z_4-Linear Goethals Codes*. Proceedings of the Third International Conference on Digital Inforation Processing and Communications, UAE, pp. 114-121, (Dubai 2013).

[46] R. E. Kalman, *Contribution to the theory of optimal control*. Boletín de de la Sociedad Matemática Mexicana. Vol 5, pp.102-119, (1960).

[47] S. Kullback, Information Theory and Statistics. Dover, Reprint of 1959 edition published by Wiley, (New York 1968).

[48] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications. Prentice-Hall*, Englewoods Cliffs, NJ, (1983).

[49] S. Lin, *Introduction to Error Correcting Codes*. Prentice-Hall, Englewood Cliffs, NJ, (1970).

[50] R.J. McEliece, *The algebraic theory of convolutional codes*, in Handbook of Coding Theory, V. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier Science, vol. 1, pp. 1065-1138, (1998).

[51] D. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge university Press, chapter 48, pp.576-583, (2003).

[52] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, vol. 16 of North-Holland Mathematical Library. North-Holland, ninth edition, (1996).

[53] D. Mandelbaum, *Decoding beyond the Designed Distance of Certain Algebraic Codes*, Info. and Control, vol. 35, pp. 209-228, (1977).

[54] J.L. Massey, *Shift-Register Synthesis and BCH Decoding*, IEEE Transactions on Information Theory, vol. IT-15, N1, (January 1969).

[55] J.L. Massey, D.J. Costello, and J. Justesen. *Polynomial weights and code constructions*, IEEE Trans. Inform. Theory, IT-19(1), pp 101-110, (1973).

[56] J.L. Massey, M.K. Sain. *Codes, automata, and continuous systems: Explicit interconnections*. IEEE Trans. Automat. Contr., AC-12(6), pp 644-650, (1967).

[57] M.O. Medeni and E.M. Souidi, *A Novel Steganographic Protocol from Error-correcting Codes*. Journal of Information Hiding and Multimedia Signal Processing, vol.1, 2010.

[58] C. Munuera, *Steganography From A Coding Theory Point Of View*. Department of Applied Mathematics, University of Valladolid.

[59] NASA Tech Briefs, *Technical Support Package for Tutorial on Reed-Solomon Error Correction Coding*, Lyndon B. Johnson Space Center, Houston, Texas

[60] W.W. Peterson, *Error-Correcting Codes*. Cambridge, Mass: M.I.T. Press, and New York: Wiley, ch.9, (1961).

[61] Ph. Piret, *Convolutional codes. An algebraic Approach*, The MIT Press, Cambridge, Massachussets, (1988).

[62] V. Pless and Z. Qian, *Cyclic codes and quadratic residue codes over* $\mathbb{Z}_4$, IEEE Transactions on Information Theory, vol. 42, N5, pp. 1594-1600, (1996).

[63] D. Radkova , D.J. Van Zanten, *Constacyclic codes as invariant subspaces*, Linear Algebra and its Applications 430, pp. 855-864, (2009).

[64] D. Radkova, A. Bojilov, A.J. Van Zanten, *Cyclic codes and quasi-twisted codes: an algebraic approach*, Report MICC 07-08, Universiteit Maastricht (2007).

[65] I.S. Reed, *A class of multiple errors correcting codes and the decoding scheme*, IRE Trans. Inform. Theory IT-4, pp. 38-49, (1954).

[66] J. Rosenthal, *Connections between linear systems and convolutional codes*, Springer, (2000)

[67] J. Rosenthal, *An algebraic Decoding Algorithm for Convolutional Codes*, Progress in Systems and Control Theory, Vol. 25 © Birkhauser Verlag Basel/Switzerland, (1999)

[68] J. Rosenthal, R. Smarandache, *Maximum distance separable convolutional codes*, Applicable algebra in engineering, Commun. Comput. 10 (1999) pp. 15-32.

[69] J. Rosenthal, J. Schumacher, E.V. York, *On behaviors and convolutional codes*, IEEE Trans. Inform. Theory 42 (6) (1996) 1881-1891.

[70] J. Rosenthal, E.V. York, *BCH convolutional codes*, IEEE Trans. Inform. Theory 45 (6) (1999) pp. 1833-1844.

[71] J. Rosenthal, E.V. York, *On Behaviors and Convolutional Codes*, IEEE Trans. Inform. Theory 42 (6) (1996) pp. 1881-1891

[72] J. Rosenthal, R. Smarandache, V. Tomás, *Decoding of Convolutional Codes Over the Erasure Channel*, IEEE Trans. Inform. Theory 58 (1) (2012).

[73] R. Roth, *Introduction to Coding Theory*, Cambridge University Press (2006) pp. 26-31.

[74] A. Salagean, *Factoring Polynomials over Z_4 and over certain Galois rings*, Finite Fields and their Applications, Vol. 11, Issue 1, (January 2005), pp. 56-70.

[75] E. C. Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal, Vol. 27, pp. 379-423, (July 1948).

[76] D. Slepian, editor. *Key Papers in the Development of Information Theory.* IEEE Press, New York, (1973)

[77] R. Smarandache, H. Gluesing-Luerssen, and J. Rosenthal, *Generalized first order descriptions and canonical forms for convolutional codes.* In Proceedings of the MTNS, Padova, Italy, (1998)

[78] P. Solé: *A quaternary cyclic code, and a family of quadriphase sequences with low correlation properties.* Lecture Notes in Computer Science 388, pp. 193-201, (1989).

[79] A. Sridharan, Design and Analysis Of LDPC Convolutional Codes, Graduate Program in Electrical Engineering, Notre Dame, Indiana. (February 2005).

[80] M. Sudan. *List Decoding: Algorithms and Applications.* In IFIP TCS, vol. 1872, Lecture Notes in Computer Science, (2000)

[81] S. Sundaram, *Linear systems* Lecture Notes in Electrical and Computer Engineering, University of Waterloo, Canada. ()

[82] R. M. Tanner, A recursive approach to low complexity codes. IEEE Trans. Inform. Theory 27, N. 5, pp. 533-547, (1983).

[83] L.E. Um, El M. Souidi, M.I. Garcia-Planas. *Error correcting codes under linear systems point of view.* Electronic IPACS Library. (2011).

[84] A. J. Viterbi, *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE Transactions on Information Theory, **13**, (2), pp. 260-269, (1967).

[85] N. Wiberg, H. A. Loeliger and R. Koetter, Codes and Iterative Decoding on General Graphs. European Trans. on Telecommunications 6, N 5, pp. 513-525, (1995).

[86] E. J. Weldon, Jr. and W. W. Peterson. Error Correcting Codes. MIT Press, Cambridge, Mass.. Second Ed, 1971.

[87] X. Wang and S. B. Wicker, "A soft output decoding algorithm for concatenated systems", IEEE Transactions on Information Theory, vol. 42, N2, pp. 543 - 553 (1996).

[88] A. Westfeld, *F5 Steganographic Algorithm*, Proc. Of the Information Hiding 4^{th} International Workshop, vol. 2137, pp. 289-302, 2001.

[89] M. Wood, *Convolutional Codes Over Rings*, Department of Electrical and Computer Engineering, Queens University, Canada (2009).

[90] M. Wood, *Convolutional Codes Over Rings*, Department of Electrical and Computer Engineering, Queens University, Canada (2009).

[91] Ch. K. Wu, Ed Dawson, *Existence of Generalized Inverse of Linear Transformations over Finite Fields*. Finite Fields and Their Applications. **4**, (4), pp. 307-315, (1998).

[92] R. W. Yeung, A first course in information theory. Springer, New York, pp. 1-5, (2002).

[93] R. W. Yeung, R. Li Shuo-Yen, N. Cai, Z. Zhang, Network Coding Theory, Foundation and Trends® in Communications and Information Theory, vol 2, nos 4 and 5, pp. 1-4, (2006).

List of publications

1. M. I. García-Planas, El M. Souidi, **L.E. Um**, *Convolutional codes under linear systems point of view. Analysis of output-controllability.* Wseas Transactions on Mathematics. Vol. 11 (4), (2010), pp. 324-333.

2. **L.E. Um**, El M. Souidi, M.I. Garcia-Planas. *Error correcting codes under linear systems point of view.* Electronic IPACS Library. (2011).

3. M. I. Garcia-Planas, El M. Soudi, **L. E. Um**. *Analysis of control properties of concatenated convolutional codes.* Cybernetics and Physics. Vol. 1 (4), (2012), pp. 252-257

4. M. I. Garcia-Planas, El M. Souidi, **L. E. Um**. *Convolutional codes under control theory point of view. Analysis of output-observability.* Recent Advances in Circuits, Communications & Signal Processing, (2013), pp. 131-137.

5. M. I. Garcia-Planas, El M. Souidi, **L. E. Um**. *Concatenated convolutional Codes. Analysis of control properties under linear systems theory point of view.* 3^{eme} Edition Des Journées Nationales de la Sécurité(JNS3), DOI: 10.1109/JNS3.2013.6595475. PP. 1–6 , (2013), IEEE Xplore Digital Library(2013).

6. M. I. Garcia-Planas, El M. Souidi, **L. E. Um**. *Decoding Algorithm for Convolutional Codes under Linear Systems Point of View.* Recent Advances in Circuits, Systems, Signal Processing and Communications, (2014), pp. 17-24.

7. J. L. Domínguez-García, M. I. García-Planas, **L. E. Um**. *Sufficient conditions for controllability of serial concatenated linear systems.* Advances in Applied and Pure Mathematics. (2014), pp. 123-127.

8. M.I. Garcia-Planas, S. Tarragona, **L.E. Um**. *Códigos de convolución desde el punto de vista de teoría de control. Análisis de la observabilidad*. Ciber, Revista Hispánica de Tendencias en Ciberseguridad. **1**, (1), (2014). pp. 1-8.

9. M.I. Garcia-Planas, J.L. Domínguez, L.E. Um. *Sufficient conditions for controllability and observability of serial and parallel concatenated linear systems*. International journal of circuits, systems and signal processing. Vol. 8, (2014) pp. 622-630.

List of Communications

1. 5th International Scientific Conference on Physics and Control (Physcon-2011), September 5-8, 2011. León Spain.
   Error correcting codes under linear systems point of view.

2. 3rd International Conference on Multimedia Computing and Systems (ICMCS'12), May 10-12, 2012. Tangier, Morocco. IEEE/SAI co-sponsored Conference.
   Properties of convolutional codes under linear systems point of view. A Survey.

3. 3rd edition of the National Security Days (JNS3), April 26-27, 2013. ENSIAS, Rabat, Morocco.
   Concatenated convolutional Codes. Analysis of control properties under linear systems theory point of view.

4. 8th International Conference on Circuits, Systems, Signal and Telecommunications (CSST'14), January 10-12, 2014. Tenerife, Spain.
   Decoding Algorithm for Convolutional Codes under Linear Systems Point of View.

5. 2nd International Conference on Mathematical, Computational and Statistical Sciences (MCSS'14). May 15-17, 2014. Gdansk, Poland.
   Sufficient conditions for controllability of serial concatenated linear systems.

6. Algebra, Codes and Networks (ACN 2014), June 16-20, 2014. Université de Bordeaux, France.
   Decoding of convolutional codes under linear systems.