UCLouvain

École polytechnique de Louvain

Graph signal processing for time-varying signals issued from linear dynamical systems

Author: **Elise DARCHE**
Supervisor: **Jean-Charles DELVENNE**
Readers: **Jean-Charles DELVENNE, Laurent JACQUES, Raphaël JUNGERS, Michaël FANUEL**
Academic year 2019–2020
Master [120] in Mathematical Engineering

Abstract

In a world where the amount of data around us has never been so important, collecting data residing on graph structures is an important issue. However, it is often impossible to collect all the data from every node on a graph. In this thesis, we focus on the problem of sampling and reconstruction of time-varying graph signals issued from linear dynamical systems.
In the first part, we review the sampling and reconstruction problem for arbitrary static and time-varying graph signals. To solve this problem, the sampling theory uses prior knowledge about the underlying graph structure through the graph shift (adjacency matrix or graph Laplacian) which is one of the building blocks of Graph Signal Processing. No prior knowledge about the underlying dynamic of these signals is usually taken into account.
In the second part, we propose a new graph shift which exploits the knowledge we have when signals are issued from linear dynamical systems. We show that this allows to systematically have a sparse representation of the signal in the frequency domain and to put a bound on the sufficient number of nodes to sample for perfect recovery of the whole signal. We find that it improves the performances obtained in sampling and in reconstruction and that using this new graph shift stays efficient in case of noisy or nonlinear signals. Building a dynamical graph based on the dynamic of the signal, we also show how we can identify a valid sampling set by applying some coloring rules. Finally, we show by numerical results how the stability of the linear system has an impact on the optimal sampling strategy to apply.

Acknowledgements

I would first like to express my sincere gratitude to my supervisor Professor Jean-Charles Delvenne for his constant availability and for his support during this year. The insightful conversations we had greatly helped me for the realization of this work.

I would also like to thank Michaël Fanuel, Professor Laurent Jacques and Professor Raphaël Jungers for accepting to spend time in reading my thesis.

Thanks also to all my friends for supporting me and for making these years of study the most beautiful ones.

Last but not the least, I would like to thank my family: my parents and my sister for their unconditional support throughout my years of study and through the process of researching and writing this master thesis.

Contents

Introduction

We live in a world where the amount of data around us has never been so important. Every aspect of our lives can produce data which is then being collected and processed. From our interactions to our consumer habits, from our displacements to our health data, everything has an utility, everything can be recorded. These data have the particularity that they reside on complex and irregular structures. A powerful tool to model these complex structures is graph theory. We can think of many examples such as the Facebook graph where each node is a user and friendships between users are modeled by edges between nodes. Above these graph structures, we can add values to the different nodes of a network. This is then called a graph signal. We could for example look at the average wage in a social network or we could be interested in the spread of a virus across the cities of the world and thus look at the number of infected people on a global transportation network. The spread of a virus is a typical example of a signal on a graph that would vary with time.

Collecting data on graphs is an important issue but it is often impossible to collect all the data from every node on a graph. Unlike in classical signal processing, the underlying network where the signal resides can bring us information about the signal through its structure. A natural question then arises : how can we use this information to know which data we should collect in order to have the best reconstruction of the whole signal. This question has already been extensively studied in the literature in the case of static and of time-varying signals. However, only the prior knowledge about the graph structure is commonly used.

In this document, we will study in more details how we can adapt the classical theory to take advantage of some prior knowledge about the signal itself.

Summary

In the first part, we will look at arbitrary signals defined on the nodes of a network and, more precisely, we will be interested in time-varying graph signals.

The first chapter reviews the basic concepts of graph signal processing and the notions of sampling and reconstruction for static signals. The second chapter builds up on this and reviews a sampling strategy that minimizes the effect of noise on the

reconstruction error. Two algorithms from the literature are described that approximately solve this problem.

In the third chapter, the notions of the first chapter are adapted to a two-dimensional framework allowing to model time-varying graph signals. These notions have already been well studied in many articles and naturally derive from the one-dimensional case. In the next chapter, we adapt the sampling strategy from the second chapter to the spatiotemporal framework and we show how the two algorithms presented can also be used in this case. We discuss about the emerging issues when sampling a time-varying graph signal.

The second part adapts the classical graph signal processing theory to the case where signals are issued from linear dynamical systems.

In the fifth chapter, we look at signals that satisfy a periodic constraint and adapt the classical theory to this class of signals. For such signals, the dynamic of the system can bring us a lot of information and we propose another representation for the underlying network which exploits the underlying dynamic of the signal. We then show how we can identify a valid set of nodes that allow us to reconstruct the whole signal from this dynamic.

The next chapter adapts these notions to signals that are non-periodic. Again, we can gain more insights about the signal by looking at the underlying dynamic of the system and we propose yet another representation for the underlying network which exploits the underlying dynamic of the signal. In the same manner as in the periodic case, we show how we can identify a valid set of nodes from this dynamic and we also show how the stability of the system can have an impact on the sampling strategy to adopt.

We conclude this thesis by applying in the last chapter the notions developed to a signal modeling the spread of a disease across the United States. The underlying network is build from airports transportation data and the signal is modeled by the compartmental SIR model commonly used in epidemiology. We compare the performances of the representation we introduced in the previous chapter to one of the classical representation used in the literature.

Contribution

Our main contribution is a new representation of the underlying graph structure for sampling graph signals issued from linear dynamical systems. We also provide a characterization of a valid sampling set for such signals.

Part I

Graph signal processing for arbitrary signals

Chapter 1

Graph signal processing : sampling theory

Graph signal processing (GSP) extends the well-known theory of discrete signal processing (DSP) which takes place in the 'time' domain to signals that reside on irregular structures from the 'space' domain represented by networks. In this chapter, we review some basic concepts of graph signal processing before focusing on the task of sampling a graph signal which has already been well-studied.

1.1 Graph signal processing

In this section, we will briefly review the basic concepts of graph signal processing (GSP). [Ortega et al., 2018] [Sandryhaila and Moura, 2014b]

We will consider a (directed or undirected) network $G = (V, E)$ with $V = \{v_0, ..., v_{N-1}\}$ the set of nodes and E the set of edges.

1.1.1 Graph shift

In classical discrete signal processing, the filter z^{-1} is defined as the *time shift*. Finite time signals are then represented as finite degree polynomials in z^{-1}. This concept of shift can be extended to graph signals x that are present on the nodes of an arbitrary graph $G = (V, E)$ instead of taking place in the time domain. A graph signal is defined as a signal whose samples (in the complex space $\mathbb{C}$) are indexed by the nodes of the graph. Formally, we have $x : V \to \mathbb{C}^N$ such that $x = [x_0, ..., x_{N-1}]^T \in \mathbb{C}^N$ where the i^{th} value x_i corresponds to the i^{th} node v_i.

In the time domain, we build various filters as polynomials in z^{-1}. We extend this notion to a graph filter h which is represented by a function $\boldsymbol{H}(\cdot) : \mathbb{C}^N \to \mathbb{C}^N$. Looking at linear filters, we can represent them by a matrix $\boldsymbol{H}$ and the filtering of a graph signal $x \in \mathbb{C}^N$ can be written as a matrix-vector multiplication : $x_{out} = \boldsymbol{H}x$.

In analogy with the signal processing theory, we want to build a *graph shift* operator $\boldsymbol{A}$ from which other filters are built. Then, we can represent a graph G algebraically using this matrix called the *graph shift*. Several matrices can be used as the graph shift operator and we will see in the next sections that this choice has an impact on the sampling operation performed on the graph. The adjacency matrix $\boldsymbol{A}_{adj}$ is convenient for directed and undirected graphs and as we will see in section 1.1.3, is consistent with the classical discrete signal processing theory. It operates on the signal x by replacing the signal value x_n at node v_n by a linear combination of the values at the neighbors of v_n. Another matrix that can be used is the graph Laplacian which only applies to undirected graphs but then has the advantages that it is symmetric and positive semi-definite. The graph Laplacian is defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}_{adj}$ where $\boldsymbol{D}$ is the degree matrix. It operates on the signal x by replacing the signal value x_n at node v_n by the difference between its own value and the values at the neighbors of v_n.

In what follows, we consider linear shift-invariant graph filters. A filter represented by $\boldsymbol{H}$ is shift-invariant if it commutes with the graph shift, $\boldsymbol{AH} = \boldsymbol{HA}$. Let us note that all linear shift-invariant graph filters are polynomials of the graph shift $\boldsymbol{A}$, i.e. $\boldsymbol{H} = h(\boldsymbol{A}) = \sum_{m=0}^{M-1} h_m \boldsymbol{A}^m$. An important property that is used in section 1.1.2 to derive the graph Fourier transform is that the eigenvectors of the shift operator $\boldsymbol{A}$ are the eigenfunctions of the polynomial filter $\boldsymbol{H}$. Indeed, let $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ be the spectral decomposition of the graph shift $\boldsymbol{A}$, we then have

$$\begin{aligned}
\boldsymbol{H} &= h(\boldsymbol{A}) \\
&= h(\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}) \\
&= \sum_{m=0}^{M-1} h_m (\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1})^m \\
&= \boldsymbol{V} h(\boldsymbol{\Lambda}) \boldsymbol{V}^{-1}
\end{aligned}$$

and we can then show that

$$\begin{aligned}
\boldsymbol{H} v_m &= \boldsymbol{V} h(\boldsymbol{\Lambda}) \boldsymbol{V}^{-1} v_m \\
&= \boldsymbol{V} h(\boldsymbol{\Lambda}) e_m \\
&= h(\lambda_m) v_m
\end{aligned}$$

In what follows, the graph shift is always denoted by the letter $\boldsymbol{A}$ which should not be confused with the adjacency matrix denoted by $\boldsymbol{A}_{adj}$.

1.1.2 Graph Fourier transform

One of the main tool of GSP is the graph Fourier transform which is the expansion of a graph signal x in the frequency domain by using basis elements that are invari-

ant to filtering. Since all filters built from $\boldsymbol{A}$ have the same eigenvectors as $\boldsymbol{A}$, we can take as basis elements the eigenbasis of the graph shift $\boldsymbol{A}$ (or the Jordan eigenbasis of $\boldsymbol{A}$ if it is not diagonalizable). For simplicity, we will assume that $\boldsymbol{A}$ has a complete eigenbasis (the results can easily be extended in the case where it does not).

Thus, let $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ be the spectral decomposition of $\boldsymbol{A}$ with $\boldsymbol{\Lambda} \in \mathbb{C}^{N\times N}$ the diagonal matrix of the eigenvalues sorted in ascending order and $\boldsymbol{V}$ the matrix of the corresponding eigenvectors. From this decomposition, the graph Fourier transform $\hat{x}$ of $x \in \mathbb{C}^N$ is defined as the decomposition of the signal in this Fourier basis :

$$\hat{x} = \boldsymbol{V}^{-1}x \tag{1.1}$$

The inverse Fourier transform is

$$x = \boldsymbol{V}\hat{x} \tag{1.2}$$

The frequency content of the signal x is represented by $\hat{x}$.

1.1.3 Discrete Fourier transform

Let us look at a special case where the graph Fourier transform is applied on a signal situated on a ring graph of T nodes. This corresponds to the situation of a discrete time periodic signal of period T. When using the adjacency matrix as the graph shift, we get the following spectral decomposition

$$\boldsymbol{A} = \boldsymbol{A}_{adj} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$$

with

$$\boldsymbol{\Lambda} = \operatorname{diag}\begin{bmatrix} e^{\frac{-2\pi 0}{T}} & e^{\frac{-2\pi}{T}} & \dots & e^{\frac{-2\pi(T-1)}{T}} \end{bmatrix}$$

and

$$\boldsymbol{V} = \frac{1}{\sqrt{T}}\left[\omega_T^{kn}\right]$$

with $\omega_N = e^{-j\frac{2\pi}{T}}$ and $k, n = 0, ..., T-1$.

Hence, we see that the eigenvector matrix is the discrete Fourier matrix and the graph Fourier transform of a time signal x gives exactly the classic discrete Fourier transform when using the adjacency matrix as graph shift on a ring graph. As was said earlier, we see that the adjacency matrix as graph shift is consistent with the discrete Fourier transform.

1.1.4 Graph frequencies

Let us now add a few words about the ordering of graph frequencies [Ortega et al., 2018] [Sandryhaila and Moura, 2014b] [Sandryhaila and Moura, 2014a].

In the time domain, the concepts of high and low frequencies are directly related to the values of the frequencies which are the eigenvalues of the cyclic shift as we have just seen above, i.e.

$$\Omega_k = \frac{2\pi k}{T}, k = 0, ..., T-1.$$

We can relate these frequencies to the degree of the variation of the spectral components as follows : the lowest frequency corresponds to the least varying spectral component and the degree of variation increases with increasing frequencies.

In GSP, this direct correspondence between the ordered value of the frequency and the corresponding degree of variation of the spectral component is less intuitive. In the time domain, we can define the total variation of a discrete signal s as the cumulative magnitude of the signal change over time :

$$\mathrm{TV}(s) = \sum_n |s_n - s_{n-1}| \tag{1.3}$$

Similarly, when using the adjacency matrix as the graph shift, we can define the total variation on graphs as the measure of the difference between the signal samples at each vertex and at its neighbors on the graph, i.e. a measure of the similarity between a graph signal and its shifted version :

Definition 1.1.1. *(Total variation on graphs)* [Sandryhaila and Moura, 2014b] The total variation on a graph G of a graph signal x is defined as

$$\mathrm{TV}_G(x) = ||x - \boldsymbol{A}_{adj}^{\mathrm{norm}} x||_1 \tag{1.4}$$

where $\boldsymbol{A}_{adj}^{\mathrm{norm}} = \frac{1}{|\lambda_{max}|}\boldsymbol{A}_{adj}$.

From this definition, we can now order the graph frequency components in the order of increasing variation. Low frequencies will correspond to frequency components with smaller variations and high frequencies will correspond to frequency components with higher variations as in the classical discrete signal processing theory.

Theorem 1.1.1. *[Sandryhaila and Moura, 2014b] Consider two distinct real eigenvalues $\lambda_m, \lambda_n \in \mathbb{R}$ of the adjacency matrix $\boldsymbol{A}_{adj}$ with corresponding eigenvectors v_m and v_n. If the eigenvalues are ordered as*

$$\lambda_m < \lambda_n$$

then the total variations of their eigenvectors satisfy

$$TV_G(v_m) > TV_G(v_n)$$

Proof. From the definition of the total variation (1.4), it follows that

$$\begin{aligned}
\mathrm{TV}_G(v) &= ||v - \boldsymbol{A}_{adj}^{\mathrm{norm}} v||_1 \\
&= \left|\left| v - \frac{1}{|\lambda_{max}|} \boldsymbol{A}_{adj} v \right|\right|_1 \\
&= \left|\left| v - \frac{\lambda}{|\lambda_{max}|} v \right|\right|_1 \\
&= \left| 1 - \frac{\lambda}{\lambda_{max}} \right| ||v||_1
\end{aligned}$$

with $(v, \lambda) = (v_m, \lambda_m)$ or (v_n, λ_n). Since we can scale v_m and v_n to have the same ℓ_1-norm and from the ordering of λ_m and λ_n, the difference between the total variations of the two eigenvectors satisfies

$$\begin{aligned}
\mathrm{TV}_G(v_m) - \mathrm{TV}_G(v_n) &= \left| 1 - \frac{\lambda_m}{|\lambda_{max}|} \right| - \left| 1 - \frac{\lambda_n}{|\lambda_{max}|} \right| \\
&= \left(1 - \frac{\lambda_m}{|\lambda_{max}|} \right) - \left(1 - \frac{\lambda_n}{|\lambda_{max}|} \right) \\
&= \frac{\lambda_n - \lambda_m}{|\lambda_{max}|} > 0,
\end{aligned}$$

which yields $\mathrm{TV}_G(v_m) > \mathrm{TV}_G(v_n)$. □

Thus, from this theorem, if a graph has a real spectrum and its frequencies are ordered as

$$\lambda_0 > \lambda_1 > ... > \lambda_{N-1}$$

then λ_0 represents the lowest frequency and λ_{N-1} represents the highest frequency. For graphs with complex spectra, frequencies are ordered by their distance from the point $|\lambda_{max}|$ on the complex plane, where λ_{max} is the eigenvalue with the largest magnitude. Figure 1.1 illustrates this result. We can see that the eigenvector associated to the largest eigenvalue varies the least and has a small range of different values (between -0.5 and -0.2) while the eigenvector associated to the smallest eigenvalue varies the most (large value differences between neighboring nodes) with a large range of different values (between -0.65 and 0.4).

A similar result can be derived when using the Laplacian matrix as the graph shift instead of the adjacency matrix. Since $\boldsymbol{L}$ is positive semi-definite, its eigenvalues are all real and non-negative ($0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{N-1}$) and it has a complete

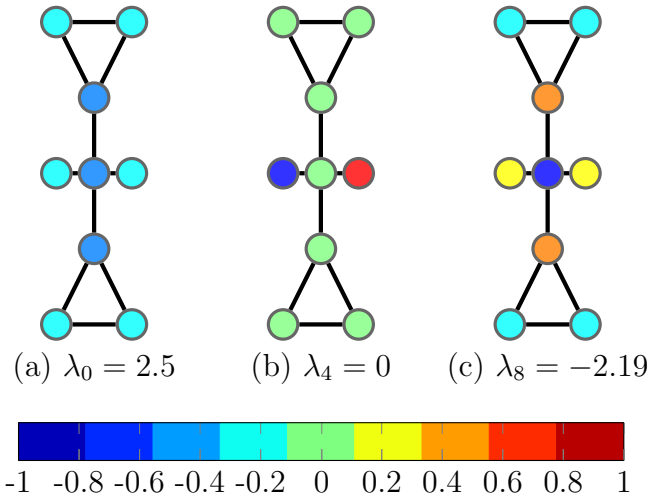


Figure 1.1: Elementary frequencies obtained from the adjacency matrix shift operator

set of orthonormal eigenvectors v_n, $n = 0, ..., N-1$. In this case, the total variation is defined as

$$\mathrm{TV}_L(x) = \sum_{n=0}^{N-1} \left(\sum_{m \in \mathcal{N}_n} \boldsymbol{A}_{adj(n,m)} (x_n - x_m)^2 \right)^{1/2} = x^T \boldsymbol{L} x \tag{1.5}$$

This can be seen as a 2-norm of the variation between the value on a node and its neighbors. In particular, for an eigenvector v_n, we have

$$\mathrm{TV}_L(v_n) = \lambda_n \tag{1.6}$$

hence a small λ_n indicates a small variation while a higher one indicates a higher variation. The zero eigenvalue corresponds to a constant eigenvector. This is the least oscillating spectral component. The eigenvalues, when ordered in ascending order, correspond to spectral components that are more and more oscillating hence we obtain the result that if a graph has a real spectrum and its frequencies are ordered as

$$\lambda_0 < \lambda_1 \leq ... \leq \lambda_{N-1}$$

then λ_0 represents the lowest frequency and λ_{N-1} represents the highest frequency. This result is illustrated on Figure 1.2 with the same network as the one used with the adjacency matrix. Now, we can see that the eigenvector associated to the largest eigenvalue varies the most while the eigenvector associated to the smallest eigenvalue varies the least (it is constant).

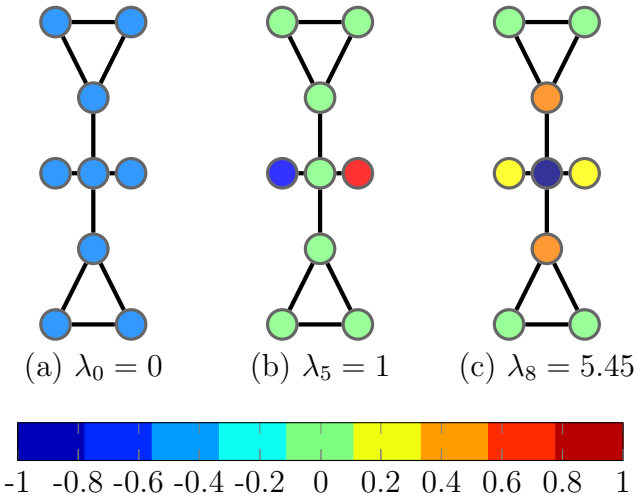


Figure 1.2: Elementary frequencies obtained from the Laplacian shift operator

1.2 Sampling on graphs

In this section, we study the problem of sampling a graph signal x and the opposite problem of recovering it from partial measurements. [Chen et al., 2015]

Suppose that we want to sample M nodes from a graph signal $x \in \mathbb{C}^N$. The sampled values are collected in $x_{\mathcal{M}} \in \mathbb{C}^M$. $\mathcal{M} = (\mathcal{M}_0, ..., \mathcal{M}_{M-1})$ denotes the sequence of sampled indices and $\mathcal{M}_i \in \{0, ..., N-1\}$ thus each signal value is indexed by one index in the vertex domain (from 0 to $N-1$). We are then interested in interpolating these measures and recover a signal x' which would ideally satisfy $x' = x$, i.e. we get perfect recovery of the original graph signal.

We define the sampling operator $\mathbf{\Psi} : \mathbb{C}^N \to \mathbb{C}^M$ as :

$$\mathbf{\Psi}_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i \\ 0, & \text{otherwise} \end{cases} \tag{1.7}$$

and the interpolation operator $\mathbf{\Phi} : \mathbb{C}^M \to \mathbb{C}^N$. Then, $\mathbf{\Psi}$ is a $M \times N$ matrix and $\mathbf{\Phi}$ is a $N \times M$ matrix.

In short,

$$\begin{cases} x_{\mathcal{M}} = \mathbf{\Psi} x \in \mathbb{C}^M \\ x' = \mathbf{\Phi} x_{\mathcal{M}} = \mathbf{\Phi}\mathbf{\Psi} x \in \mathbb{C}^N \end{cases}$$

The signal x can be recovered either exactly or approximately. In general, we have $M < N$ and $rank(\mathbf{\Phi}\mathbf{\Psi}) \leq M < N$ thus perfect recovery cannot be achieved because $\mathbf{\Phi}\mathbf{\Psi}$ is not equal to the identity matrix. Of course, perfect recovery is always possible when $M = N$ (then $\mathbf{\Phi}\mathbf{\Psi} = \boldsymbol{I}$) but we are not interested in this case. It has

been shown that perfect recovery with $M < N$ is possible for signals that have a certain structure.

Let us define this special class of signals as bandlimited graph signals.

Definition 1.2.1. A graph signal $x \in \mathbb{C}^N$ is called K-bandlimited if there is a $K \in \{0, 1, ..., N-1\}$ such that its graph Fourier transform $\hat{x}$ satisfies

$$\hat{x}_k = 0 \text{ for all } k \geq K$$

K is called the bandwidth of x.

This condition of bandlimitedness does not necessarily mean that our signal is smooth or low-pass. Indeed, we do not take the order of the eigenvalues into account so we can reorder them (and permute the corresponding eigenvectors) to choose any band. However, if we sort the eigenvalues of the graph shift in a descending order for the adjacency matrix or in an ascending order for the graph Laplacian, then the bandlimited graph signal is indeed smooth (as seen in 1.1.2). Thus in what follows, when we talk about bandlimitedness, we only talk about a limit on the number of non-zero signal coefficients in the graph Fourier domain. This allows to be more general and consider signals that are not necessarily smooth as well.

Recall the eigendecomposition of the graph shift $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$. Let us now denote as $\boldsymbol{V}_{(K)} \in \mathbb{R}^{N \times K}$ the matrix $\boldsymbol{V}$ where we only select the K columns corresponding to the non-zero coefficients in the graph Fourier domain. The following theorem allows us to recover a K-bandlimited graph signal using only K samples.

Theorem 1.2.1 (Rank theorem)**.** *[Chen et al., 2015] Let $\boldsymbol{\Psi}$ from equation* (1.7) *satisfy*

$$rank(\boldsymbol{\Psi}\boldsymbol{V}_{(K)}) = K \tag{1.8}$$

with $\boldsymbol{V}_{(K)}$ as explained above and with K the bandwidth. For all graph signals x with bandwidth K, perfect recovery, $x = \boldsymbol{\Phi}\boldsymbol{\Psi}x$, is achieved by choosing

$$\boldsymbol{\Phi} = \boldsymbol{V}_{(K)}\boldsymbol{U} \tag{1.9}$$

with $\boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{V}_{(K)}$ a $K \times K$ identity matrix.

This theorem tells us first that we should sample at least K nodes so the sample size M should be at least equal to the bandwidth K. Let us note that this value is a lower bound. When $M < K$, the matrix $\boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{V}_{(K)}$ can never be an identity matrix. When $M = K$, $\boldsymbol{U}$ is the inverse of $\boldsymbol{\Psi}\boldsymbol{V}_{(K)}$ and it is a left inverse of $\boldsymbol{\Psi}\boldsymbol{V}_{(K)}$ when $M > K$. In that case, we will take $\boldsymbol{U}$ as the pseudo-inverse of $\boldsymbol{\Psi}\boldsymbol{V}_{(K)}$. Sampling more nodes than necessary can be useful to reduce the influence of noise for example.
The second conclusion to draw from this theorem is that we should choose the sampled nodes such that $rank(\boldsymbol{\Psi}\boldsymbol{V}_{(K)}) = K$. This is always possible when $M \geq K$.

Indeed, the matrix $\boldsymbol{V}$ is invertible hence its columns are linearly independent. It follows that $rank(\boldsymbol{V}_{(K)}) = K$ always holds and there always is a set of K linearly independent rows in $\boldsymbol{V}_{(K)}$.
When the sampling operator $\boldsymbol{\Psi}$ satisfies (1.8), we then have what we call a qualified sampling operator.

Chapter 2

Sampling algorithms

In section 1.2, we saw that in order to get perfect recovery of a sampled signal, we need a qualified sampling operator meeting the conditions stated in theorem 1.2.1, i.e. when having a signal x that is K-bandlimited, one should design a sampling operator $\boldsymbol{\Psi}$ such that $rank(\boldsymbol{\Psi}\boldsymbol{V}_{(K)}) = K$ with $\boldsymbol{V}_{(K)}$ the eigenvector matrix of the graph shift where we select the K columns corresponding to the non-zero coefficients of x in the graph Fourier domain. To design such a qualified sampling operator for a K-bandlimited signal x, we need to find K linearly independent rows in $\boldsymbol{V}_{(K)}$. This problem often has multiple solutions. In this section, an approach is proposed to design an optimal sampling operator that minimizes the effect of noise [Chen et al., 2015] and two algorithms are described that provide some good approximations to the optimal sampling operator [Chen et al., 2015] [Tzamarias et al., 2018].

2.1 Optimal sampling operator

In this section, we show how to design an optimal sampling operator in the sense that it minimizes the effect of noise. We then describe two algorithms that provide some good approximations to the optimal sampling operator. Other algorithms than the ones provided here exist such as random sampling [Puy et al., 2018] [Varma and Kovačević, 2019]. These algorithms are computationally more efficient than the ones presented in this section but provide worse results in terms of the reconstruction error [Tzamarias et al., 2018] and will not be presented here.

In what follows, we consider that we have a graph signal x that is K-bandlimited on a simple graph G with graph shift $\boldsymbol{A}$. As said in section 1.2, when having the matrix $\boldsymbol{V}_{(K)}$ from $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, a qualified sampling operator $\boldsymbol{\Psi}$ should select K linearly-independent rows, which is always possible. When there are multiple choices, we could choose to design a qualified sampling operator that minimizes the effect of noise [Chen et al., 2015].

Let us consider a sampling operation where noise e is added during the sampling operation as follows :

$$x_{\mathcal{M}} = \mathbf{\Psi} x + e$$

The reconstruction process then yields

$$x' = \mathbf{\Phi} x_{\mathcal{M}} = \mathbf{\Phi}\mathbf{\Psi} x + \mathbf{\Phi} e = x + \mathbf{\Phi} e$$

and the reconstruction error can be bounded by

$$\begin{aligned} ||x' - x||_2 &= ||\mathbf{\Phi} e||_2 = ||\boldsymbol{V}_{(K)}\boldsymbol{U} e||_2 \\ &\leq ||\boldsymbol{V}_{(K)}||_2 ||\boldsymbol{U}||_2 ||e||_2 \end{aligned}$$

With $||\boldsymbol{V}_{(K)}||_2$ and $||e||_2$ fixed, we want to minimize the spectral norm of $\boldsymbol{U}$. From theorem 1.2.1, we know that $\boldsymbol{U}$ is obtained from the inverse or pseudo-inverse of $\mathbf{\Psi}\boldsymbol{V}_{(K)}$. Minimizing the spectral norm of $\boldsymbol{U}$ is thus equivalent to maximizing the smallest singular value of $\mathbf{\Psi}\boldsymbol{V}_{(K)}$,

$$\mathbf{\Psi}^{opt} = \arg\max_{\mathbf{\Psi}} \sigma_{\min}(\mathbf{\Psi}\boldsymbol{V}_{(K)}) \tag{2.1}$$

where $\sigma_{\min}$ denotes the smallest singular value. We call $\mathbf{\Psi}^{opt}$ an optimal sampling operator.

Remark. A problem that naturally arises when dealing with signals is the case where the signal is not exactly bandlimited. We defined the concept of a bandlimited signal in Definition 1.2.1. Let us note that these requirements are restrictive and few signals in real-world applications are effectively bandlimited. A second class of signals can then be defined where the requirements are relaxed [Chen et al., 2016].

Definition 2.1.1. A graph signal $x \in \mathbb{C}^N$ is approximately bandlimited with parameters $\beta \geq 1$ and $\mu \geq 0$, if there is a $K \in \{0, 1, ..., N-1\}$ such that its graph Fourier transform $\hat{x}$ satisfies

$$\sum_{k=K}^{N-1} (1 + k^{2\beta}) \hat{x}_k^2 \leq \mu ||x||_2^2$$

.

This definition states that most of the energy (2-norm) of the signal is captured by K frequencies but it allows for other frequencies to capture the remaining energy. Dealing with this kind of signals, we could design an optimal sampling operator that would minimize the effect of the error when sampling K nodes of the graph where the approximately bandlimited signal x (with approximate bandwidth K) resides. Minimax lower bounds on the recovery error under different sampling strategies are derived in [Chen et al., 2016] as well as a recovery strategy for this kind of signals. We will not study this in further details here. In what follows, we concentrate on the recovery strategy presented in theorem 1.2.1 and on the sampling strategy presented in this section.

2.1.1 Greedy algorithm

The problem of finding $\mathbf{\Psi}^{opt}$ is claimed to be NP hard in [Chen et al., 2015]. To approximately solve (2.1), we can use a greedy algorithm presented in Algorithm 1 which has been shown to give a good approximation to the global optimum [Avron and Boutsidis, 2013]. In what follows, let us recall that the notation $\mathcal{M}$ indicates the sampling sequence and $(\boldsymbol{V}_{(K)})_{\mathcal{M}}$ denotes the sampled rows from $\boldsymbol{V}_{(K)}$.

Algorithm 1: Optimal Sampling Operator via Greedy Algorithm

Input : $V_{(K)}$ the K columns of V corresponding to the nonzero Fourier coefficients, M the number of samples
Output : $\mathcal{M}$ sampling set

1: **while** $|\mathcal{M}| < M$ **do**
2: $\quad m = \arg\max_i \sigma_{\min}((V_{(K)})_{\mathcal{M}+\{i\}})$
3: $\quad \mathcal{M} \leftarrow \mathcal{M} + \{m\}$
4: **end while**
5: **return** $\mathcal{M}$

This method has to find and store M eigenvectors to sample M nodes from a graph of N nodes and the time complexity of the sampling algorithm is of $\mathcal{O}(N|\mathcal{M}|^4) = \mathcal{O}(NM^4)$ [Tzamarias et al., 2018] [Avron and Boutsidis, 2013]. Indeed, at each iteration m, we have to compute the singular value decomposition of $(\boldsymbol{V}_{(K)})_{\mathcal{M}+\{i\}}$ for each remaining node i, yielding a complexity of $\sum_{m=1}^{M}(N-m+1)\mathcal{O}(M^3) = \mathcal{O}(NM^4)$.

Let us note that this algorithm returns one sampling set while there may sometimes be multiple sampling sets that are equally optimal. Indeed, at line 2, there may be multiple nodes i that, when added to the sampling set $\mathcal{M}$, give the same minimum singular value $\sigma_{\min}((\boldsymbol{V}_{(K)})_{\mathcal{M}+\{i\}})$. In this case, the chosen node depends on the implementation of the algorithm. This particularly happens when the graph presents some symmetries.

2.1.2 Iterative algorithm

In the previous section, a greedy algorithm was presented which provides a good approximation to the optimal sampling operator (2.1). In this section, we present another sampling method which is also developed for K-bandlimited graph signals and which provides similar results as Algorithm 1 concerning the reconstruction error while having a better computational complexity. [Tzamarias et al., 2018]

We start by noting that Algorithm 1 is not scalable when the size of the matrix $\boldsymbol{V}_{(K)}$ and the number of required samples M increase. Indeed, it requires a singular value decomposition at each iteration and this increases the complexity.

In [Tzamarias et al., 2018], an iterative sampling algorithm was proposed which achieves a better trade-off between the computational complexity and the reconstruction error. This algorithm relies on theorem 1.2.1. It finds K linearly independent rows from the matrix $\boldsymbol{V}_{(K)}$ in an iterative way. It starts with an empty set and at each iteration, adds one node to the sampling set $\mathcal{M}$ such that the sampling set always selects $|\mathcal{M}|$ linearly independent rows at each iteration. This is presented in Algorithm 2.

Algorithm 2: Optimal Sampling Operator via Recursive Algorithm

Input : $V_{(K)}$ the K columns of V corresponding to the nonzero Fourier coefficients, M the number of samples
Output : $\mathcal{M}$ sampling set

1: $\mathcal{M} \leftarrow i$, where i is the index of any nonzero element of the first eigenvector v_1
2: **for** $m = 2$ to M **do**
3: $\quad x \in \mathrm{Ker}((V_{(m)})_{\mathcal{M}}),\ x \neq 0$
4: $\quad b = (V_{(m)})_{\mathcal{M}^c} x$
5: $\quad i \leftarrow \arg\max_i |b_i|$
6: $\quad \mathcal{M} \leftarrow \mathcal{M} + \mathcal{M}^c(i)$
7: **end for**
8: **return** $\mathcal{M}$

Let us analyze the different steps of the algorithm. The same notations as in Algorithm 1 are adopted, namely $(\boldsymbol{V}_{(K)})_{\mathcal{M}}$ denotes the sampled rows from $\boldsymbol{V}_{(K)}$.

The first node added in the sampling set $\mathcal{M}$ is a node with row index corresponding to any nonzero element of the first eigenvector in $\boldsymbol{V}_{(K)}$ (line 1). Then at each iteration m, we build the $m \times (m+1)$ matrix $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m}$ which has a rank equal to m since $(\boldsymbol{V}_{(m)})_{\mathcal{M}_m}$ has full rank. The nullspace of this matrix gives the unique unit vector x that is orthogonal to all rows of $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m}$ (line 3). We then look for the row in $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m^c}$ that is the 'most linearly dependent' to x. $\mathcal{M}_m^c$ indicates the complement of $\mathcal{M}_m$. To that end, we look at the absolute maximum element index i of the product vector $b = (\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m^c} x$ (line 5). This gives the index of the most linearly dependent row to x, i.e., the row that is the most aligned with x, and the i^{th} row of $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m^c}$ is linearly independent from all rows of $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_m}$. The vertex corresponding to this row is added to the sampling set $\mathcal{M}_m$ and we obtain $\mathcal{M}_{m+1}$ (line 6). The matrix $(\boldsymbol{V}_{(m+1)})_{\mathcal{M}_{m+1}}$ is now of rank $m+1$ and we can continue to iterate until $|\mathcal{M}| = M$.

Again, as was the case for the first algorithm, there may sometimes be multiple nodes i that give the same maximum value for $|b_i|$ at line 5 and in that case, the chosen node again depends on the implementation of the algorithm. Algorithm 2

hence returns only one sampling set while there may sometimes exist other sampling sets that are as good as the one returned.

It has been shown in [Tzamarias et al., 2018] that Algorithms 1 and 2 provide similar results in terms of the reconstruction error. Both methods have to store M eigenvectors to sample M nodes from a graph of N nodes. However, the complexity of the sampling algorithm is of $\mathcal{O}(NM^4)$ for the first method and of $\mathcal{O}(M^4 + NM^2)$ for the second one [Tzamarias et al., 2018]. The second method is thus more efficient compared to the first one when searching for the sampling set and this is particularly true for large graphs.

Chapter 3

Graph signal processing on product graphs

So far, we have presented the sampling theory concerning graph signals that were static in time. However, many real-world applications involve signals that, while residing on a static graph structure, also evolves with time. In [Sandryhaila and Moura, 2014a], it was proposed to represent the two-dimensional space-time domain where time-varying graph signals reside by taking the Cartesian product between the space domain and the time domain. The theory for simple graph signals then naturally extends to the case of time-varying graph signals residing on these two-dimensional graph structures.

3.1 Discrete signal processing on product graphs

In this section, we will extend the concepts seen in section 1.1 to product graphs and more particularly to the space-time framework.

A product graph between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined as

$$G = G_1 \diamond G_2 = (V, E)$$

with $|V_1| = N_1$, $|V_2| = N_2$ and $|V| = N_1 N_2$. Different product graphs exist, the most studied being the Kronecker, Cartesian and strong product [Sandryhaila and Moura, 2014a]. In what follows, we will concentrate on the Cartesian graph product because it arises naturally when representing a spatiotemporal graph.

We can think of a spatiotemporal graph as a simple graph that has been replicated multiple times. Let us consider a (directed or undirected) network $G_S = (V_S, E_S)$ with $V_S = \{v_0, ..., v_{N-1}\}$ the set of nodes. From this spatial network, we build a larger network by replicating it T times where T is the number of time steps. If the time network is an undirected path or cycle, each node is then linked to its

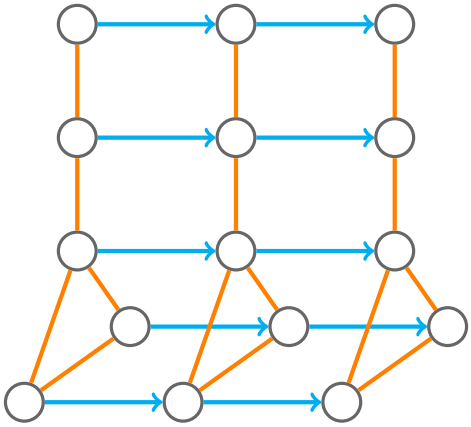


Figure 3.1: Spatiotemporal graph. The spatial links are in orange and the temporal links are the blue arrows.

spatial neighbors but also to itself in the future and in the past in an undirected way. If the time network is a directed path or cycle, each node is then linked to its spatial neighbors and there is a directed link between the node and its future self. An example of such a spatiotemporal graph is shown on Figure 3.1 where we have used a directed path as time graph.

This can be seen as making the Cartesian product between the spatial network G_S and a temporal graph $G_T = (V_T, E_T)$ where $V_T = \{t_0, ..., t_{T-1}\}$. As mentioned, the temporal graph can be a (directed or undirected) ring graph as well as a path (directed or undirected). For the Cartesian product graph, denoted as $G_\times = G_T \times G_S$, the adjacency matrix is defined as

$$\boldsymbol{A}_{adj,\times} = \boldsymbol{A}_{adj,T} \oplus \boldsymbol{A}_{adj,S} \triangleq \boldsymbol{A}_{adj,T} \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes \boldsymbol{A}_{adj,S} \in \mathbb{R}^{NT \times NT}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. $\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker product between matrices $\boldsymbol{A} = [a_{m,n}] \in \mathbb{C}^{M \times N}$ and $\boldsymbol{B} \in \mathbb{C}^{K \times L}$ which is a $KM \times LN$ matrix with block structure

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{0,0}\boldsymbol{B} & \dots & a_{0,N-1}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{M-1,0}\boldsymbol{B} & \dots & a_{M-1,N-1}\boldsymbol{B} \end{bmatrix}.$$

$\boldsymbol{A} \oplus \boldsymbol{B}$ denotes the Kronecker sum defined as above. The graph Laplacian of the product graph is similarly defined as $\boldsymbol{L}_\times = \boldsymbol{L}_T \oplus \boldsymbol{L}_S$ since we have

$$\begin{aligned} \boldsymbol{L}_\times &= \boldsymbol{D}_\times - \boldsymbol{A}_{adj,\times} \\ &= \boldsymbol{D}_T \oplus \boldsymbol{D}_S - \boldsymbol{A}_{adj,T} \oplus \boldsymbol{A}_{adj,S} \\ &= (\boldsymbol{D}_T \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes \boldsymbol{D}_S) - (\boldsymbol{A}_{adj,T} \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes \boldsymbol{A}_{adj,S}) \\ &= (\boldsymbol{D}_T - \boldsymbol{A}_{adj,T}) \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes (\boldsymbol{D}_S - \boldsymbol{A}_{adj,S}) \\ &= \boldsymbol{L}_T \oplus \boldsymbol{L}_S \end{aligned}$$

Instead of studying simple graph signals, we will now consider time-varying graph signals. We can represent a time-varying graph signal as a matrix $X \in \mathbb{C}^{N \times T}$ where the i^{th} row corresponds to the temporal signal on the i^{th} node and the j^{th} column corresponds to the spatial or graph signal at the j^{th} time instant. We here see that using a ring as the temporal graph would imply a periodic time-varying graph signal.

3.1.1 Joint Fourier transform

Let us generalize the graph Fourier transfrom (GFT) and the discrete Fourier transform (DFT) seen in sections 1.1.2 and 1.1.3 to our time-varying graph signal $X \in \mathbb{C}^{N \times T}$. [Loukas and Foucard, 2016]

Let $\boldsymbol{A}_S = \boldsymbol{V}_S \boldsymbol{\Lambda}_S \boldsymbol{V}_S^{-1}$ and $\boldsymbol{A}_T = \boldsymbol{V}_T \boldsymbol{\Lambda}_T \boldsymbol{V}_T^{-1}$ be the eigendecompositions of the graph shifts of G_S and G_T respectively. We can apply the graph Fourier transform on each column of X (each column is a spatial signal at a particular time instant) which gives

$$\text{GFT}\{X\} = \boldsymbol{V}_S^{-1} X. \tag{3.1}$$

By doing this, we take into account the variation of the signal with respect to the graph but we neglect the temporal aspect of the data. We can also apply the discrete Fourier transform on each row of X (each row being a time signal on one node of the graph) which gives

$$\text{DFT}\{X\} = X \boldsymbol{V}_T^{-T}. \tag{3.2}$$

In this case, we neglect the graph structure of the data. Let us note here that we call the transformation given by (3.2) 'Discrete Fourier Transform' or DFT to indicate that we look at the frequency content of the time-varying graph signal X in the *time frequency domain* (as opposed to the GFT where we look at the frequency content in the *graph frequency domain*). The notation DFT does not necessarily mean that we use a ring graph as temporal graph as it was done in section 1.1.3.

Now, applying these two transformations jointly, we are then able to represent the frequency content of the time-varying graph signal X along both time and graph domains. This leads us to the definition of the joint graph and time Fourier transform given by

$$\text{JFT}\{X\} = \boldsymbol{V}_S^{-1} X \boldsymbol{V}_T^{-T}. \tag{3.3}$$

Let us define $x = \text{vec}(X)$ as the signal X where the columns have been stacked on top of one another. Using the formula $(\boldsymbol{B}^T \otimes \boldsymbol{A})\text{vec}(X) = \text{vec}(\boldsymbol{A}X\boldsymbol{B})$, we can also write (3.3) in matrix-vector notation as

$$\text{JFT}\{x\} = (\boldsymbol{V}_T^{-1} \otimes \boldsymbol{V}_S^{-1})x. \tag{3.4}$$

Equation (3.4) can also be obtained by applying the classical graph Fourier transform on the Cartesian product graph $G_\times$. To show this, we need the following lemma.

Lemma 3.1.1. *[Kurokawa et al., 2017] Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (resp. $\boldsymbol{B} \in \mathbb{R}^{m\times m}$) have linearly independent eigenvectors $u_1, ..., u_n$ (resp. $v_1, ..., v_m$) and corresponding eigenvalues λ_i for $i = 1, ..., n$ (resp. μ_j for $j = 1, ..., m$). Then the set of Kronecker products $\{v_j \otimes u_i\}_{i=1,...,n;j=1,...,m}$ are linearly independent eigenvectors of $\boldsymbol{B} \oplus \boldsymbol{A}$ and correspond to eigenvalues $\lambda_i + \mu_j$.*

Proof. Let us prove that $[\boldsymbol{B} \oplus \boldsymbol{A}](v_j \otimes u_i) = (\mu_j + \lambda_i)(v_j \otimes u_i)$.

$$\begin{aligned}
[\boldsymbol{B} \oplus \boldsymbol{A}](v_j \otimes u_i) &= [\boldsymbol{B} \otimes \boldsymbol{I}_n + \boldsymbol{I}_m \otimes \boldsymbol{A}](v_j \otimes u_i) \\
&= [\boldsymbol{B} \otimes \boldsymbol{I}_n](v_j \otimes u_i) + [\boldsymbol{I}_m \otimes \boldsymbol{A}](v_j \otimes u_i) \\
&= (\boldsymbol{B}v_j \otimes u_i) + (v_j \otimes \boldsymbol{A}u_i) \\
&= (\mu_j v_j \otimes u_i) + (v_j \otimes \lambda_i u_i) \\
&= (\mu_j + \lambda_i)(v_j \otimes u_i)
\end{aligned}$$

□

Applying this result to the graph shift of $G_\times$, we obtain that

$$\begin{aligned}
\boldsymbol{A}_\times &= \boldsymbol{A}_T \oplus \boldsymbol{A}_S \\
&= (\boldsymbol{V}_T \otimes \boldsymbol{V}_S)(\boldsymbol{\Lambda}_T \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes \boldsymbol{\Lambda}_S)(\boldsymbol{V}_T \otimes \boldsymbol{V}_S)^{-1} \\
&= \boldsymbol{V}_\times \boldsymbol{\Lambda}_\times \boldsymbol{V}_\times^{-1}
\end{aligned}$$

with $\boldsymbol{V}_\times = \boldsymbol{V}_T \otimes \boldsymbol{V}_S$. Thus, we can compute the joint Fourier transform of the time-varying graph signal X as the graph Fourier transform of the graph signal $x = \mathrm{vec}(X)$ as

$$\mathrm{GFT}\{x\} = \boldsymbol{V}_\times^{-1} x = (\boldsymbol{V}_T^{-1} \otimes \boldsymbol{V}_S^{-1})x \tag{3.5}$$

which yields the same result as above.

Figure 3.2 summarizes the relationships between the joint, discrete and graph Fourier transforms [Loukas and Foucard, 2016]. A directed arrow from one transform to the other indicates that the first one can be simulated by the second one. It was already explained in section 1.1.3 how, when taking a ring graph $\mathcal{T}$, the GFT could simulate the DFT. The DFT and the GFT are both special cases of the JFT when taking respectively a graph with a single vertex ($N = 1$) and a graph where the signal does not vary in time ($T = 1$). Finally, equation (3.5) exactly shows how we can simulate the JFT with the GFT by considering the joint graph $\mathcal{J}$ as one large graph and the time-varying signal X as a simple graph signal $x = \mathrm{vec}(X)$.

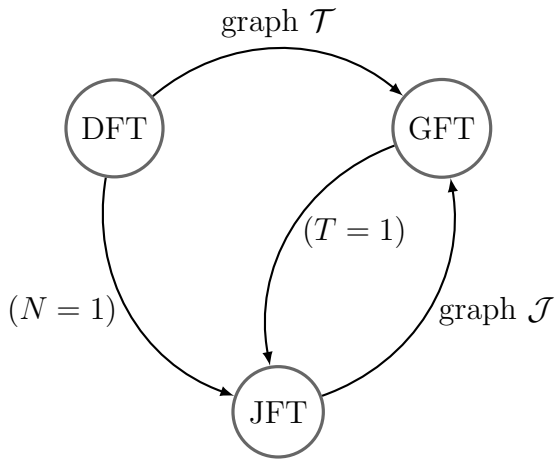


Figure 3.2: Relationships between the different Fourier transforms.

3.1.2 Joint graph frequencies

We can also extend the notions of graph frequencies seen in section 1.1.4 in the case of product graphs. However, a way to look at the product graph frequencies is also to look at the time frequencies and the space frequencies separately. Indeed, since we have

$$\begin{aligned}\boldsymbol{A}_{\times} &= \boldsymbol{A}_T \oplus \boldsymbol{A}_S \\ &= (\boldsymbol{V}_T \otimes \boldsymbol{V}_S)(\boldsymbol{\Lambda}_T \otimes \boldsymbol{I}_N + \boldsymbol{I}_T \otimes \boldsymbol{\Lambda}_S)(\boldsymbol{V}_T \otimes \boldsymbol{V}_S)^{-1},\end{aligned}$$

each joint frequency is given by $\lambda = \lambda_t + \lambda_s$. We can gain more insight by looking at the frequencies separately. Indeed, a low joint frequency will be the result of a low time and a low space frequency and this is also the case with high frequencies which are the the result of two high frequencies in time and space. But concerning the frequencies in the middle of the spectrum, these can be the result of a small time frequency and a high space frequency, or the result of two middle frequencies, or the result of a high time frequency and a low space frequency etc. Hence looking at the frequencies separately gives more information about the signal than looking at them jointly. A signal that varies slowly across time but that varies a lot in the space domain will excite more the low time frequencies and the high space frequencies. A signal that varies slowly both in space and in time will excite more the low frequencies both in time and space etc. We can analyze graph signals in terms of the frequency characteristics along each factor graph instead of one unique joint dimension [Kurokawa et al., 2017].

3.2 Sampling on product graphs

In this section, we study the problem of sampling a time-varying graph signal X and the opposite problem of recovering it from partial measurements. We will extend

the results obtained in section 1.2 to the spatiotemporal case.

The problem of sampling a time-varying graph signal can be naturally derived from the problem of sampling a graph signal as follows. Suppose that we want to sample J nodes from a time-varying graph signal $X \in \mathbb{C}^{N \times T}$. The sampled values are collected in $X_{\mathcal{ML}}$. $\mathcal{ML} = ((\mathcal{M}_0, \mathcal{L}_0), ..., (\mathcal{M}_{J-1}, \mathcal{L}_{J-1}))$ denotes the sequence of sampled indices and $\mathcal{M}_i \in \{0, ..., N-1\}$, $\mathcal{L}_i \in \{0, ..., T-1\}$ thus each signal value is indexed by one index in the vertex domain (from 0 to $N-1$) and one index in the time domain (from 0 to $T-1$). We are then interested in interpolating these measures and recover a signal X' which would ideally satisfy $X' = X$, i.e. we get perfect recovery of the original time-varying graph signal.

Working in a spatiotemporal (two-dimensional) framework, we can distinguish two different manners of tackling this problem.
First of all, we can look at the joint frequency content of the signal X on the whole product graph by computing the joint Fourier transform from (3.3) or (3.4). Then, if the signal is bandlimited in the joint Fourier domain, we can consider the spatiotemporal (or product) graph as one complex and large graph and the time-varying signal as a simple graph signal, i.e. we do not look at time and space separately and work in a one-dimensional framework. This problem is equivalent to the problem of sampling a simple graph signal as seen in section 1.2. This will be referred to as the case where space and time are coupled together, meaning that we do not look at them separately.
Secondly, we can try and exploit the particular structure of our product graph and consider sampling in space and time separately. In this section, we will study in more details this case and see when it is interesting to apply this strategy.

3.2.1 Sampling in space and time separately

Let us analyze the case where we take into account the fact that the spatiotemporal graph is indeed a Cartesian product graph and we consider the signal X as it is, i.e. a time-varying graph signal. Let us recall that $X \in \mathbb{C}^{N \times T}$ where each column of X is a graph signal at one time instant and each row of X is a time signal on one particular node.

In the case where we want to exploit the structure of the product graph, we should look at the time frequency content of each time signal (each row of X) on the temporal graph G_T by computing the discrete Fourier transform from (3.2) and we should also look at the graph frequency content of each graph signal (each column of X) on the space graph G_S by computing the graph Fourier transform from (3.1). The frequency band of the signal in the time dimension is the union of all the time frequencies where at least one of the time signals has a non-zero frequency component, i.e. the union of the frequency bands of each time signal. Similarly, the frequency

band of the signal in the space dimension is the union of all the space frequencies where at least one of the graph signals has a non-zero frequency component, i.e. the union of the frequency bands of each graph signal. We can consider the two-dimensional frequency band of the whole time-varying graph signal X as being the Cartesian product between its frequency band in the time domain and its frequency band in the space domain. The two-dimensional frequency band of X is then a space-by-time rectangle.

Let us consider that the signal X is K-bandlimited in the space domain and K'-bandlimited in the time domain. The frequency band of the signal is a K-by-K' rectangle and this implies that each graph signal (columns of X) on G_S is at most K-bandlimited on the same set of K spatial frequencies at any time t and that each temporal signal (rows of X) on G_T is at most K'-bandlimited on the same set of K' time frequencies at any node. Then we can get perfect recovery of the signal by sampling $J = ML \geq KK'$ samples. Indeed, if we have a set of K nodes $\mathcal{M} = \{v_0, ..., v_{K-1}\} \in V_S$ that is a valid (i.e., satisfying the rank theorem 1.2.1) set of sampling nodes at any time and a set of K' nodes $\mathcal{L} = \{t_0, ..., t_{K'-1}\} \in V_T$ that is a valid set of sampling time at every spatial node, then the Cartesian product $\{v_0, ..., v_{K-1}\} \times \{t_0, ..., t_{K'-1}\}$ is a valid set of spatiotemporal sampling for $G_\times$.

We thus define $\mathbf{\Psi}_S \in \mathbb{C}^{M\times N} : \mathbb{C}^{N\times T} \to \mathbb{C}^{M\times T}$ as

$$\mathbf{\Psi}_S(i,j) = \begin{cases} 1, & j = \mathcal{M}_i \\ 0, & \text{otherwise} \end{cases}$$

and $\mathbf{\Psi}_T \in \mathbb{C}^{L\times T} : \mathbb{C}^{N\times T} \to \mathbb{C}^{N\times L}$ as

$$\mathbf{\Psi}_T(i,j) = \begin{cases} 1, & j = \mathcal{L}_i \\ 0, & \text{otherwise} \end{cases}$$

and the interpolation operators $\mathbf{\Phi}_S \in \mathbb{C}^{N\times M} : \mathbb{C}^{M\times T} \to \mathbb{C}^{N\times T}$ and $\mathbf{\Phi}_T \in \mathbb{C}^{T\times L} : \mathbb{C}^{N\times L} \to \mathbb{C}^{N\times T}$.

In short,

$$\begin{cases} X_{\mathcal{ML}} = \mathbf{\Psi}_S X \mathbf{\Psi}_T^T \in \mathbb{C}^{M\times L} \\ X' = \mathbf{\Phi}_S X_{\mathcal{ML}} \mathbf{\Phi}_T^T = \mathbf{\Phi}_S \mathbf{\Psi}_S X \mathbf{\Psi}_T^T \mathbf{\Phi}_T^T \in \mathbb{C}^{N\times T} \end{cases}$$

As before, we have to choose $\mathbf{\Psi}_S$ and $\mathbf{\Psi}_T$ such that $rank(\mathbf{\Psi}_S \boldsymbol{V}_{S(K)}) = K$ and $rank(\mathbf{\Psi}_T \boldsymbol{V}_{T(K')}) = K'$ where $\boldsymbol{V}_S$ and $\boldsymbol{V}_T$ are the eigenvector matrices of the graph shift of the spatial and temporal graph respectively. We also have to choose $\mathbf{\Phi}_S$ and $\mathbf{\Phi}_T$ according to theorem 1.2.1.

Let us note that it is always possible to include the frequency band of X in a rectangle. In the worst case, we can take an N-by-T rectangle which includes all

frequencies but this would be sub-optimal since we would have to sample all nodes from the product graph. Here, we are interested in the case where this rectangle can be made smaller than that, i.e. when some space frequencies are not excited by any of the graph signals (columns of X) and when some time frequencies are not excited by any of the time signals (rows of X).

Let us conclude this section on a reflection that will be deepened later. When sampling a time-varying graph signal X on a spatiotemporal graph $G_\times$, multiple cases can arise. First, the signal X could be bandlimited on the whole product graph but not separately in time and in space. This would mean that the frequency content of X is zero on some joint frequencies when computing the joint Fourier transform (3.3) but that the space-by-time rectangle should be of size N-by-T in order to include all non-zero frequency components of X. It is then more optimal (in terms of the number of nodes to sample) to consider a sampling strategy based on the notions presented in section 1.2. Second, the signal X could be bandlimited both in time and space separately (with bandwidth that may have different width) and hence also on the whole product graph. In this case, we can consider a sampling strategy based on the notions presented in this section, i.e. we sample X in time and in space separately and we take the Cartesian product of these samples, or we can consider a sampling strategy based on the notions presented in section 1.2. We can then choose the one that is the more practical or that yields the lowest sampling set size for example. We will see examples of these two cases later.

Chapter 4

Sampling algorithms on product graphs

In chapter 3, the sampling theory of graph signals was extended to the case of time-varying graph signals. We have seen that it is possible to deal with such two-dimensional signals in two different manners. We can either consider time and space jointly. The sampling theory is then equivalent to the one-dimensional case and we get a sampling set on the joint product graph. Or we can consider time and space separately and then work with the Cartesian product of the sampling set in time and the sampling set in space. We now show how we can extend the notion of the optimal sampling operator minimizing the effect of noise from section 2.1 to the two-dimensional case. We will see that it is possible to design an optimal space-time sampling operator by designing separately two sampling operators, one in time and one in space. Consequently, we can apply both the greedy algorithm and the iterative algorithm from sections 2.1.1 and 2.1.2 separately in time and space. We conclude this chapter with some toy examples to illustrate the notions seen so far .

4.1 Optimal sampling operator

In this section, we show how the optimal sampling operator from section 2.1 can be adapted to the case where we sample on a spatiotemporal graph (or more generally on a product graph). We describe how Algorithms 1 and 2 can be used in order to sample on the product graph and we show that the method can be applied to both cases where we sample in space and time jointly or separately.

In section 2.1, it was shown that an optimal sampling operator $\mathbf{\Psi}^{opt}$ that minimizes the effect of noise on a K-bandlimited graph signal x on a simple graph G should verify

$$\mathbf{\Psi}^{opt} = \arg\max_{\mathbf{\Psi}} \sigma_{\min}(\mathbf{\Psi}\boldsymbol{V}_{(K)})$$

where $\boldsymbol{V}_{(K)}$ is the eigenvector matrix of the graph shift of G where we select the K columns corresponding to the K non-zero components of the graph Fourier transform of x.

Similarly, we can apply the same reasoning as the one developed in section 2.1 to the joint sampling of a time-varying graph signal X. We suppose that the spatial signal is K-bandlimited at each time step and that each temporal signal is K'-bandlimited at each spatial node. Let us now consider the joint sampling operation where noise E is added as follows :

$$X_{\mathcal{ML}} = \boldsymbol{\Psi}_S X \boldsymbol{\Psi}_T^T + E$$

The reconstruction process then yields

$$X' = \boldsymbol{\Phi}_S X_{\mathcal{ML}} \boldsymbol{\Phi}_T^T = \boldsymbol{\Phi}_S \boldsymbol{\Psi}_S X \boldsymbol{\Psi}_T^T \boldsymbol{\Phi}_T^T + \boldsymbol{\Phi}_S E \boldsymbol{\Phi}_T^T = X + \boldsymbol{\Phi}_S E \boldsymbol{\Phi}_T^T$$

and the reconstruction error can be bounded by

$$\begin{aligned} ||X' - X||_2 &= ||\boldsymbol{\Phi}_S E \boldsymbol{\Phi}_T^T||_2 = ||\boldsymbol{V}_{S(K)} \boldsymbol{U}_S E \boldsymbol{U}_T^T \boldsymbol{V}_{T(K')}^T||_2 \\ &\leq ||\boldsymbol{V}_{S(K)}||_2 ||\boldsymbol{U}_S||_2 ||E||_2 ||\boldsymbol{U}_T^T||_2 ||\boldsymbol{V}_{T(K')}^T||_2 \end{aligned}$$

With $||\boldsymbol{V}_{S(K)}||_2$, $||\boldsymbol{V}_{T(K')}^T||_2$ and $||E||_2$ fixed, we want to minimize the spectral norm of $\boldsymbol{U}_S$ and $\boldsymbol{U}_T^T$. As seen in section 2.1, the optimal sampling operator $\boldsymbol{\Psi}_S$ is then given by

$$\boldsymbol{\Psi}_S^{opt} = \arg\max_{\boldsymbol{\Psi}_S} \sigma_{\min}(\boldsymbol{\Psi}_S \boldsymbol{V}_{S(K)}) \tag{4.1}$$

The optimal sampling operator $\boldsymbol{\Psi}_T$ is similarly given by

$$\boldsymbol{\Psi}_T^{opt} = \arg\max_{\boldsymbol{\Psi}_T} \sigma_{\min}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')}) \tag{4.2}$$

since

$$\begin{aligned} \min_{\boldsymbol{\Psi}_T} ||\boldsymbol{U}_T^T||_2 &= \min_{\boldsymbol{\Psi}_T} ||(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')})^{-T}||_2 \\ &= \min_{\boldsymbol{\Psi}_T} \sigma_{\max}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')})^{-T} \\ &= \min_{\boldsymbol{\Psi}_T} \sigma_{\max}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')})^{-1} \\ &= \min_{\boldsymbol{\Psi}_T} \frac{1}{\sigma_{\min}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')})} \\ &= \max_{\boldsymbol{\Psi}_T} \sigma_{\min}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(K')}) \end{aligned}$$

4.1.1 Greedy algorithm

In section 2.1.1, we showed how the problem of finding $\boldsymbol{\Psi}^{opt}$ could be approximately solved by using the greedy algorithm presented in Algorithm 1. The same algorithm

can now be used separately to find an approximation to $\boldsymbol{\Psi}_S^{opt}$ and to $\boldsymbol{\Psi}_T^{opt}$.

When sampling a time-varying graph signal X, we have two possibilities. We can either check whether it is $\tilde{K}$-bandlimited on the joint Fourier domain and then apply Algorithm 1 on $\boldsymbol{V}_{(\tilde{K})}$ in order to have an optimal sampling operator, where $\boldsymbol{V}$ is the eigenvector matrix of the graph shift $\boldsymbol{A}_\times$ of the whole product graph. Or we can check if it is K-bandlimited in the graph Fourier domain and K'-bandlimited in the time Fourier domain and apply the same algorithm on both $\boldsymbol{V}_{S(K)}$ and $\boldsymbol{V}_{T(K')}$, where $\boldsymbol{V}_S$ and $\boldsymbol{V}_T$ are the eigenvector matrices of the graph shift of the space graph and of the time graph respectively. We should therefore choose to sample X in time and space separately when $KK' \leq \tilde{K}$ and in time and space jointly otherwise if we want to minimize the necessary and sufficient number of sampling nodes. Let us note that sampling in time and space separately also allows to reduce the computation load since we solve two smaller problems instead of one large one.

4.1.2 Iterative algorithm

Just as we generalized Algorithm 1 from section 2.1.1 to the case of a time-varying graph signal X on a Cartesian product graph, we can also generalize Algorithm 2 from section 2.1.2 to that case. The idea is still the same. When having a time-varying graph signal X, if the spatial signal is K-bandlimited at each time step and if each temporal signal is K'-bandlimited at each spatial node, then we can apply Algorithm 2 both to the spatial graph and to the temporal graph with the eigendecompositions of their graph shifts being $\boldsymbol{A}_S = \boldsymbol{V}_S \boldsymbol{\Lambda}_S \boldsymbol{V}_S^{-1}$ and $\boldsymbol{A}_T = \boldsymbol{V}_T \boldsymbol{\Lambda}_T \boldsymbol{V}_T^{-1}$ respectively.

4.2 Examples

In this section, we will look at various examples in order to illustrate what has been said so far. In all the examples, we consider as spatial graph G_S the five-node undirected graph from Figure 4.1 with the adjacency matrix as graph shift:

$$\boldsymbol{A}_S = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Time is represented by an undirected path with $T = 4$ nodes. We will consider the adjacency matrix as graph shift. The adjacency matrix of the Cartesian product graph is given by

$$\boldsymbol{A}_\times = \boldsymbol{A}_T \oplus \boldsymbol{A}_S$$

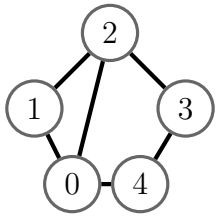


Figure 4.1: Spatial graph

The inverse graph Fourier transform matrix is

$$\boldsymbol{V}_S = \begin{bmatrix} 0.50 & -0.43 & 0.50 & 0.18 & 0.53 \\ 0 & 0.74 & 0 & 0.52 & 0.43 \\ -0.50 & -0.43 & -0.50 & 0.18 & 0.53 \\ 0.50 & 0.20 & -0.50 & -0.58 & 0.36 \\ -0.50 & 0.20 & 0.50 & -0.58 & 0.36 \end{bmatrix}$$

with frequencies

$$\boldsymbol{\Lambda}_S = \operatorname{diag}\begin{bmatrix} -2 & -1.17 & 0 & 0.69 & 2.48 \end{bmatrix},$$

the inverse discrete Fourier transform is

$$\boldsymbol{V}_T = \begin{bmatrix} 0.37 & -0.60 & -0.60 & 0.37 \\ -0.60 & 0.37 & -0.37 & 0.60 \\ 0.60 & 0.37 & 0.37 & 0.60 \\ -0.37 & -0.60 & 0.60 & 0.37 \end{bmatrix}$$

with frequencies

$$\boldsymbol{\Lambda}_T = \operatorname{diag}\begin{bmatrix} -1.62 & -0.62 & 0.62 & 1.62 \end{bmatrix},$$

and the inverse joint Fourier transform is given by the Kronecker product

$$\boldsymbol{V}_\times = \boldsymbol{V}_T \otimes \boldsymbol{V}_S \in \mathbb{R}^{20\times 20}$$

with frequencies

$$\boldsymbol{\Lambda}_\times = \boldsymbol{\Lambda}_T \oplus \boldsymbol{\Lambda}_S$$

For the sake of clarity, in each of the figures presented below representing the power spectra obtained by the discrete Fourier transform and by the graph Fourier transform, only the maximum spectrum at each time frequency and at each graph frequency respectively is shown. If none of the spectra is 0, we can then deduce that X is not bandlimited in the time and in the graph Fourier domain respectively.

Example 1

In the first example, we consider the time-varying graph signal

$$X = \begin{bmatrix} -0.3752 & -0.0232 & 0.3073 & 0.3015 \\ 0.0903 & -0.2195 & 0.7206 & 0.2194 \\ -0.3807 & 0.1660 & -0.1049 & 0.6678 \\ -0.1516 & -0.7004 & 0.0341 & -0.0108 \\ -0.5179 & -0.2882 & -0.1551 & -0.0054 \end{bmatrix},$$

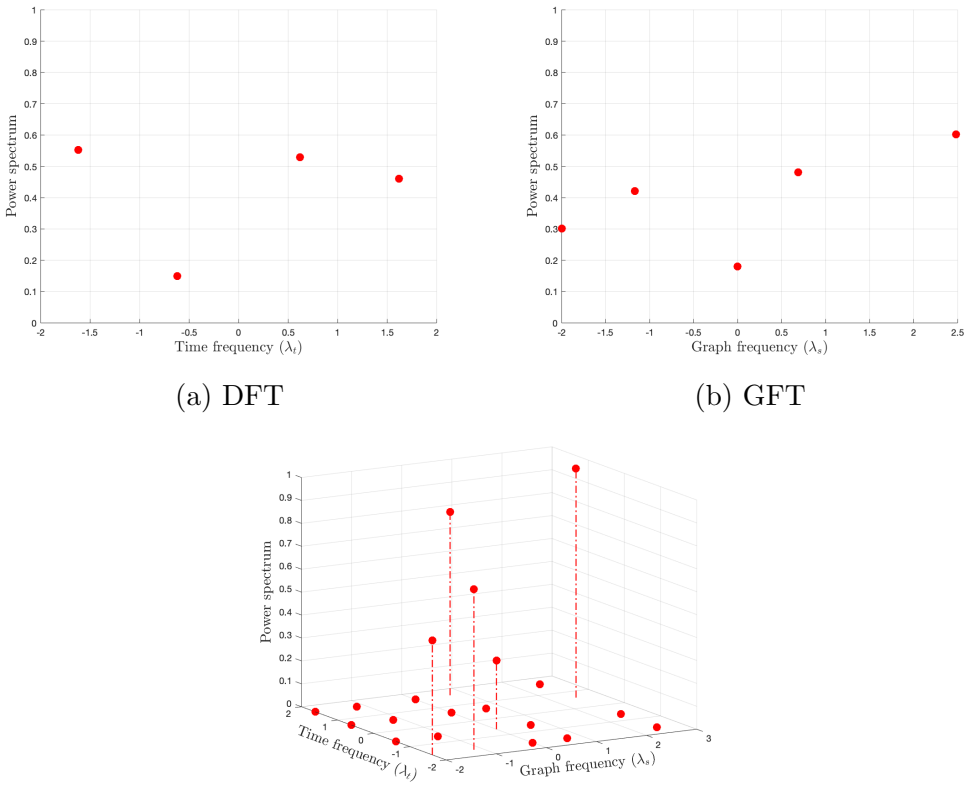
where each row is a time signal on one node and each column is a graph signal at one time instant.

Computing the discrete, graph and joint Fourier transforms, we obtain that X is neither sparse in the time Fourier domain (Figure 4.2a) nor in the graph Fourier domain (Figure 4.2b) but it is 5-bandlimited in the joint Fourier domain (Figure 4.2c). Let us recall that the notation 'DFT' indicates the time frequency domain, 'GFT' the space or graph frequency domain and 'JFT' the joint frequency domain.

We thus here have an example of a signal that is bandlimited in the joint Fourier domain but where we cannot sample in time and space separately (or we would have to sample every node). We thus have to apply one of the two algorithms seen in this section on the inverse joint Fourier transform $\boldsymbol{V}_\times$ where we only select the columns corresponding to the nonzero coefficients of the joint Fourier transform.

Applying Algorithm 1, we obtain $\mathcal{ML} = ((1,1),(2,1),(3,1),(4,1),(4,3))$ as sampling set where the first index is the index on the spatial graph and the second one is the index in time. Space and time are numbered from 0 to $N-1$ and $T-1$ respectively. We then have

$$x_{\mathcal{ML}} = \begin{bmatrix} -0.2195 & 0.1660 & -0.7004 & -0.2882 & -0.0054 \end{bmatrix}^T.$$



(a) DFT

(b) GFT

(c) JFT

Figure 4.2: Example 1 : Power spectrum of X obtained by the DFT, GFT and JFT

We recover $x = \text{vec}(X) = \mathbf{\Phi} x_{\mathcal{ML}}$ by using the interpolation operator

$$\mathbf{\Phi} = \boldsymbol{V}_{(5)}(\mathbf{\Psi}\boldsymbol{V}_{(5)})^{-1}$$
$$= \begin{bmatrix} 0.69 & 0.89 & 0.49 & 0.09 & 0.20 \\ -0.09 & 0.71 & 0.14 & -0.17 & -0.63 \\ 0.30 & -0.07 & 0.30 & 0.33 & -0.93 \\ -0.21 & 0.12 & -0.02 & 0.82 & -0.56 \\ -0.05 & 0.55 & 0.96 & -0.18 & -0.05 \\ -0.35 & 0.14 & 0.28 & -0.23 & -1.02 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.34 & -0.50 & -0.83 & 0.39 & 0.44 \\ -0.52 & 0.21 & -0.55 & -0.64 & -0.19 \\ 0.34 & -0.50 & 0.17 & -0.61 & 0.44 \\ -0.35 & 0.20 & -0.03 & 0.33 & 0.72 \\ -0.70 & -0.67 & 0.25 & 0.10 & -0.30 \\ -0.31 & -0.15 & -0.02 & -0.85 & 0.58 \\ 0.39 & -0.58 & -0.48 & -0.22 & 0.51 \\ -0.48 & -0.58 & -1.00 & 0.15 & 0.07 \\ -0.39 & -0.96 & -0.19 & 0.24 & -0.14 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

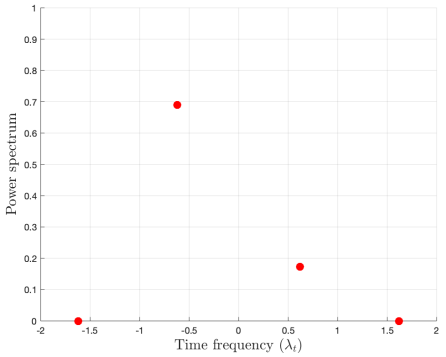
We could have applied Algorithm 2. With this algorithm, we would have sampled the nodes $\mathcal{ML} = ((0,0),(2,0),(2,1),(4,1),(0,2))$ so $\mathbf{\Psi}$ and $\mathbf{\Phi}$ would have been different but we would have proceeded in the same way concerning the computations.

Example 2

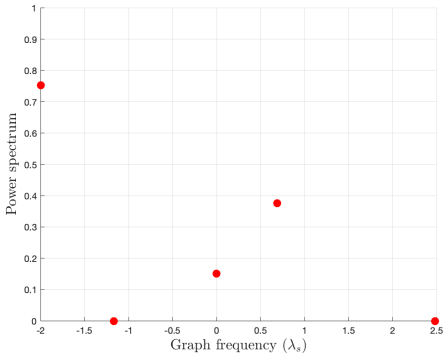
In the second example, we consider the time-varying graph signal

$$X = \begin{bmatrix} -0.5185 & 0.1923 & 0.3205 & -0.3111 \\ -0.1957 & 0.0726 & 0.1210 & -0.1174 \\ 0.3837 & -0.1423 & -0.2371 & 0.2302 \\ -0.0840 & 0.0312 & 0.0519 & -0.0504 \\ 0.5175 & -0.1919 & -0.3198 & 0.3105 \end{bmatrix}.$$

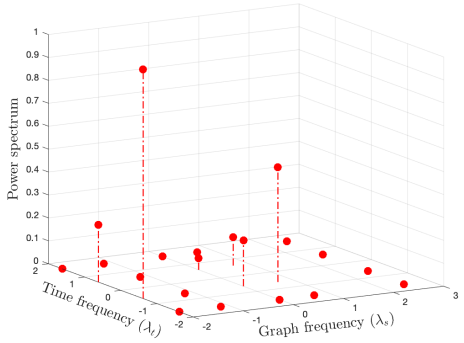
Computing the discrete, graph and joint Fourier transforms, we now obtain that X is 2-bandlimited in the time Fourier domain, 3-bandlimited in the graph Fourier domain and 6-bandlimited in the joint Fourier domain as can be seen on Figure 4.3.



(a) DFT



(b) GFT



(c) JFT

Figure 4.3: Example 2 : Power spectrum of X obtained by the DFT, GFT and JFT

In this case, it is possible to sample in time and space separately (and then take as spatiotemporal sampling set the Cartesian product between the time samples and the space samples). We can thus apply the algorithms on both $\boldsymbol{V}_S$ and $\boldsymbol{V}_T$. Since we have to sample 6 nodes in both cases (where we sample separately or not), we can ask what is the best solution. In this case, it is best to sample in time and space separately because of the complexity of the algorithms that increases with increasing size of matrices. The computational cost will be lower than when sampling jointly in time and space.

Now, applying Algorithm 1 on $\boldsymbol{V}_S$, we obtain the space sampling set $\mathcal{M} = (2, 3, 4)$ and applying the same algorithm on $\boldsymbol{V}_T$, we obtain the time sampling set $\mathcal{L} = (0, 3)$. We thus sample the nodes $\mathcal{ML} = \mathcal{M} \times \mathcal{L}$ and

$$X_{\mathcal{ML}} = \boldsymbol{\Psi}_S X \boldsymbol{\Psi}_T^T = \begin{bmatrix} 0.38 & 0.23 \\ -0.08 & -0.05 \\ 0.52 & 0.31 \end{bmatrix}^T,$$

with the sampling operators

$$\boldsymbol{\Psi}_S = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\boldsymbol{\Psi}_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We then recover X by using the interpolation operators

$$\begin{aligned} \boldsymbol{\Phi}_S &= \boldsymbol{V}_{S(3)}(\boldsymbol{\Psi}_S \boldsymbol{V}_{S(3)})^{-1} \\ &= \begin{bmatrix} -1.00 & -0.31 & -0.31 \\ 0 & -0.45 & -0.45 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{\Phi}_T &= \boldsymbol{V}_{T(2)}(\boldsymbol{\Psi}_T \boldsymbol{V}_{T(2)})^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -0.62 \\ -0.62 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Again, applying Algorithm 2, we would have sampled the nodes $\mathcal{ML} = \mathcal{M} \times \mathcal{L}$ with $\mathcal{M} = (0, 3, 4)$ and $\mathcal{L} = (0, 3)$.

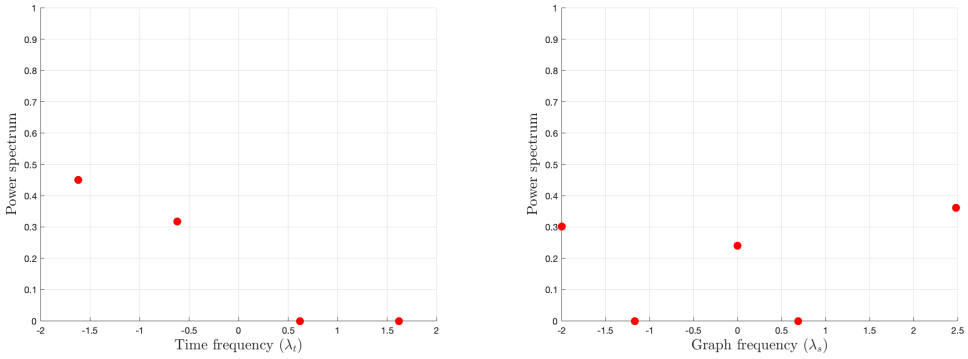
Example 3

In the third example, we consider the time-varying graph signal

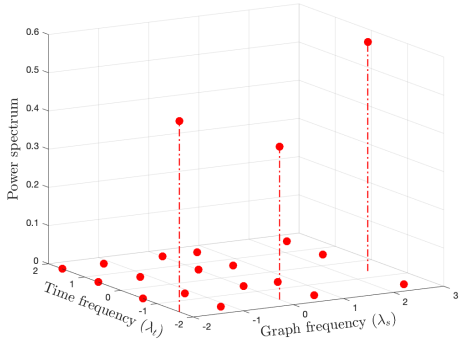
$$X = \begin{bmatrix} -0.0240 & -0.1525 & 0.3889 & -0.3585 \\ -0.1542 & 0.0953 & 0.0953 & -0.1542 \\ -0.3585 & 0.3889 & -0.1525 & -0.0240 \\ -0.1105 & 0.0497 & 0.1099 & -0.1477 \\ -0.1477 & 0.1099 & 0.0497 & -0.1105 \end{bmatrix}.$$

Computing the discrete, graph and joint Fourier transforms, we now obtain that X is 2-bandlimited in the time Fourier domain, 3-bandlimited in the graph Fourier domain and 3-bandlimited in the joint Fourier domain as can be seen on Figure 4.4.

In this case, we see that we could decide to sample in time and space separately (and



(a) DFT (b) GFT



(c) JFT

Figure 4.4: Example 3 : Power spectrum of X obtained by the DFT, GFT and JFT

then take the Cartesian product of both samplings) since the signal is bandlimited both in time and in space separately. We should then apply the algorithms on both $\boldsymbol{V}_S$ and $\boldsymbol{V}_T$ but we would have to sample $2 \cdot 3 = 6$ nodes instead of 3 if we sample in time and space jointly and apply the algorithms on $\boldsymbol{V}_\times$. Based on the complexity however, we could prefer to sample separately. Both methods will work but we will not show the reconstruction here since it is similar to what we applied above.

Part II

GSP for signals issued from linear dynamical systems

Chapter 5

Signals issued from linear dynamical systems with periodic constraints

In chapter 1 and 2, the notions of sampling theory were presented as well as two sampling algorithms. These notions concerned graph signals that were static in time. Chapter 3 and 4 extended these notions to the case of time-varying graph signals, which arise commonly in many real-world applications. The notions presented are valid for any arbitrary signal. However, it often happens that we have some prior knowledge about these time-varying signals. Common signals are the ones issued from dynamical systems. In this chapter, we explore how we can exploit this prior knowledge and more efficiently sample such signals. We study the case where signals are issued from linear dynamical systems and are periodic. The non-periodic case will be studied in the next chapter.

5.1 Sampling and reconstruction

In this section, we study the problem of sampling and reconstruction of a signal that is issued from a linear dynamical system with periodic constraints.

We consider the Cartesian product graph $G_\times = G_T \times G_S$ between our temporal graph $G_T = (V_T, E_T)$ and our spatial graph $G_S = (V_S, E_S)$. The time-varying graph signals $X \in \mathbb{C}^{N \times T}$ that we now consider are signals derived from linear dynamical systems, i.e. the columns of X satisfy

$$x_{t+1} = \boldsymbol{M} x_t \in \mathbb{C}^N \tag{5.1}$$

with $t \in \{0, ..., T-2\}$ and x_t the t^{th} column of X. Let us recall that the columns of X are the graph signals at different time instants and the rows of X are time signals

on different nodes. We add the periodic constraint that

$$x_0 = \boldsymbol{M} x_{T-1}. \tag{5.2}$$

From the constraint of periodicity, the temporal graph G_T in this section will always be a ring graph with T nodes, T being the period. Let us note that every initial condition $x_0 \in \mathbb{C}^N$ does not necessarily give a periodic signal $X \in \mathbb{C}^{N \times T}$ such that the periodic constraint (5.2) is satisfied. In the following, we note $\text{vec}(X) = x$.

Theorem 5.1.1. *Given a periodic time-varying graph signal $X \in \mathbb{C}^{N \times T}$ whose columns satisfy* (5.1) *and* (5.2)*, X is N-bandlimited at most in the joint Fourier domain of $G_\times$ when using*

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{M} & -\boldsymbol{I} & \dots & 0 & 0 \\ 0 & 0 & \boldsymbol{M} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \boldsymbol{M} & -\boldsymbol{I} \\ -\boldsymbol{I} & 0 & 0 & \dots & 0 & \boldsymbol{M} \end{bmatrix} \in \mathbb{R}^{NT \times NT} \tag{5.3}$$

as graph shift. $\tilde{\boldsymbol{M}}$ will be referred to as the periodic dynamical graph shift.

Proof. The matrix $\tilde{\boldsymbol{M}}$ is such that

$$\tilde{\boldsymbol{M}} x = 0$$

with $x = \text{vec}(X)$, hence $x \in \text{null}(\tilde{\boldsymbol{M}})$. Since all signals x such that $x \in \text{null}(\tilde{\boldsymbol{M}})$ satisfy (5.1) and (5.2), we have that $\dim \text{null}(\tilde{\boldsymbol{M}}) \leq N$. From this, we can directly deduce that x will be bandlimited with a bandwidth of at most N when taking $\tilde{\boldsymbol{M}}$ as the graph shift for $G_\times$. □

Before going into more details, we will characterize the graph shift $\tilde{\boldsymbol{M}}$ in terms of its eigenvalues and eigenvectors. Knowing the eigenvalue and eigenvector pairs (λ_n, v_n), $n \in \{1, ..., N\}$ of $\boldsymbol{M}$, we can compute the eigenvalue and eigenvector pairs of $\tilde{\boldsymbol{M}}$. Indeed, we note that $\tilde{\boldsymbol{M}}$ can be written as a Kronecker sum of two matrices, namely $\tilde{\boldsymbol{M}} = \boldsymbol{M}_T \oplus \boldsymbol{M}$ with

$$\boldsymbol{M}_T = \begin{bmatrix} 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \\ -1 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{T \times T} \tag{5.4}$$

Hence, from Lemma 3.1.1, the NT eigenvalues of $\tilde{\boldsymbol{M}}$ are

$$\tilde{\lambda}_{n,k} = -\omega_k + \lambda_n \tag{5.5}$$

with $n \in \{1, ..., N\}$, $\omega_k = e^{\frac{2\pi ik}{T}}$ and $k \in \{0, ..., T-1\}$. The eigenvectors are given by

$$\tilde{v}_{n,k} = \begin{bmatrix} v_n & \omega_k v_n & \omega_k^2 v_n & \dots & \omega_k^{T-1} v_n \end{bmatrix}. \tag{5.6}$$

Let us denote by $\mathcal{I}$ the vector space of the initial conditions $x_0 \in \mathbb{C}^N$ that allow to have a periodic signal X satisfying (5.1) and (5.2). In the next theorem, we characterize this vector space $\mathcal{I}$.

Theorem 5.1.2. *The vector space $\mathcal{I}$ of the initial conditions $x_0 \in \mathbb{C}^N$ yielding a periodic signal $X \in \mathbb{C}^{N \times T}$ of period T satisfying* (5.1) *and* (5.2) *is generated by the eigenvectors v_n of $\boldsymbol{M}$ corresponding to the eigenvalues λ_n such that λ_n is a T^{th} root of unity.*

Proof. The periodic signal $x = \text{vec}(X)$ is such that $\tilde{\boldsymbol{M}} x = 0$. Hence, from (5.5) and (5.6), we have

$$\tilde{\boldsymbol{M}} x = 0 = \sum_{n,k;\lambda_n = \omega_k} (\lambda_n - \omega_k) v_{n,k}$$

yielding

$$\tilde{\boldsymbol{M}} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{T-1} \end{bmatrix} = 0 = \sum_{n,k;\lambda_n = \omega_k} (\lambda_n - \omega_k) \begin{bmatrix} v_n \\ \omega_k v_n \\ \vdots \\ \omega_k^{T-1} v_n \end{bmatrix}$$

which means that x_0 is a linear combination of the eigenvectors of $\boldsymbol{M}$ corresponding to the zero eigenvalues $\lambda_n - \omega_k = 0$ of $\tilde{\boldsymbol{M}}$. Since we have that $\omega_k = e^{\frac{2\pi ik}{T}}$, $k \in \{0, ..., T-1\}$, are the T^{th} roots of unity, we finally get that x_0 is generated by the eigenvectors of $\boldsymbol{M}$ corresponding to the eigenvalues λ_n such that λ_n is a T^{th} root of unity. □

The next result provides a tighter bound on the bandlimitedness of a signal $X \in \mathbb{C}^{N \times T}$ satisfying (5.1) and (5.2).

Theorem 5.1.3. *Given a periodic time-varying graph signal $X \in \mathbb{C}^{N \times T}$ whose columns satisfy* (5.1) *and* (5.2)*, and given $d = dim(\mathcal{I})$, X is d-bandlimited at most in the joint Fourier domain of $G_\times$ when using $\tilde{\boldsymbol{M}}$ from* (5.3) *as graph shift.*

Proof. The matrix $\tilde{\boldsymbol{M}}$ is such that

$$\tilde{\boldsymbol{M}} x = 0$$

with $x = \text{vec}(X)$, hence $x \in \text{null}(\tilde{\boldsymbol{M}})$. Since all signals x such that $x \in \text{null}(\tilde{\boldsymbol{M}})$ satisfy (5.1) and (5.2), we have that $\dim \text{null}(\tilde{\boldsymbol{M}}) = \dim(\mathcal{I}) = d$. From this, we can directly deduce that x will be bandlimited with a bandwidth of at most d when taking $\tilde{\boldsymbol{M}}$ as the graph shift for $G_\times$. □

When representing the spectrum of the signal in the two-dimensional frequency domain, we will obtain a diagonal spectrum where time and space are in resonance with each other since the zero frequency components of X correspond to the zero eigenvalues of $\tilde{\boldsymbol{M}}$ (let us recall that $\text{vec}(X) = x \in \text{null}(\tilde{\boldsymbol{M}})$) and the eigenvalues of $\tilde{\boldsymbol{M}}$ in (5.3) are such that $-\omega_k + \lambda_n = 0$ with $-\omega_k$ the time frequencies and λ_n the space frequencies.

Example 5.1.1. Let us consider as spatial graph G_S the seven-node undirected graph from Figure 5.1.

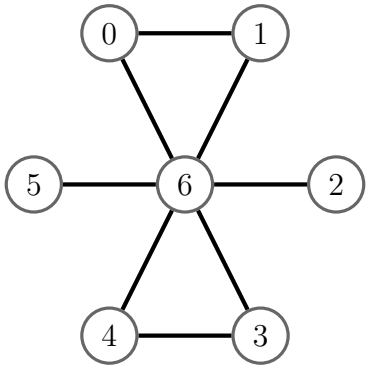


Figure 5.1: Spatial graph of example 5.1.1

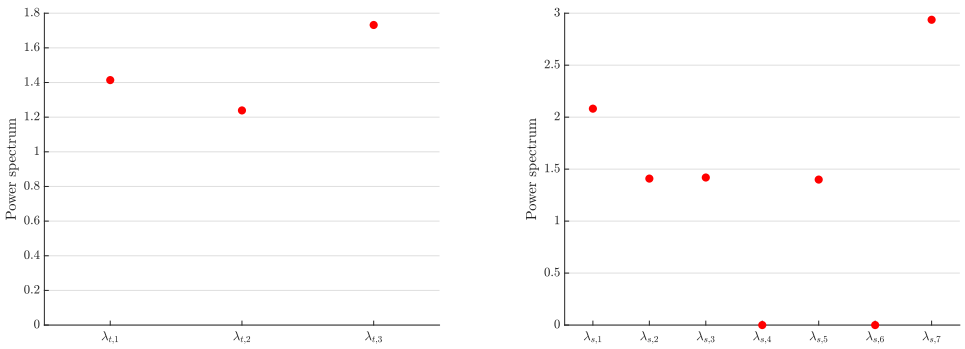
We consider a time-varying graph signal X on this graph. This signal X is such that $x_{t+1} = \boldsymbol{M} x_t$ with

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

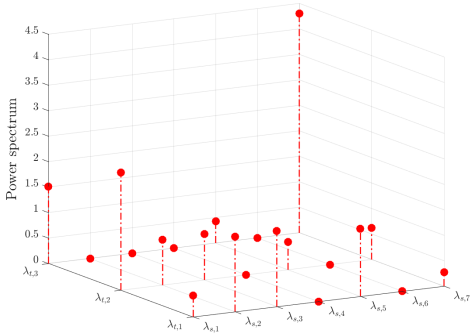
and periodic constraint $x_0 = \boldsymbol{M} x_{T-1}$. The signal X is periodic of period $T = 3$. In this case, the initial condition x_0 had to be chosen such that it was in the vector space generated by the eigenvectors of $\boldsymbol{M}$ corresponding to the eigenvalues λ_n such that λ_n is a third root of unity. The temporal graph G_T is a ring graph with $T = 3$ nodes since our signal is a periodic signal of period 3. The eigenvalues of $\boldsymbol{M}$ are $\lambda_n = \{0, 0, 0, 0, \omega_0, \omega_1, \omega_2\}$ with $\omega_j = e^{2i\pi j/T}$, $j = 0, 1, 2$, and the eigenvalues of $\boldsymbol{M}_T$ are $-\omega_k$ for $k = 0, 1, 2$.

We now look at the frequency content of the signal with the different graph shifts

at our disposal. First, taking the adjacency matrices of G_S, G_T and $G_\times$ as graph shifts, we can see on Figure 5.2 that the signal is not bandlimited in the time frequency domain (DFT), it is 5-bandlimited in the graph frequency domain (GFT) and 13-bandlimited in the joint frequency domain (JFT). We thus observe that the bound of N-bandlimitedness (stated in theorem 5.1.1) with $N = 7$ is not reached in this case when taking the adjacency matrix as graph shift.



(a) DFT (b) GFT

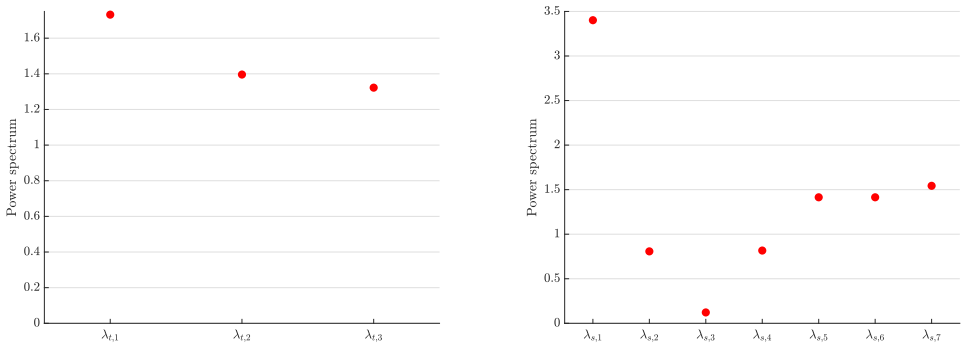


(c) JFT

Figure 5.2: Example 5.1.1 : Power spectrum of X obtained by the DFT, GFT and JFT when using the adjacency matrix as graph shift

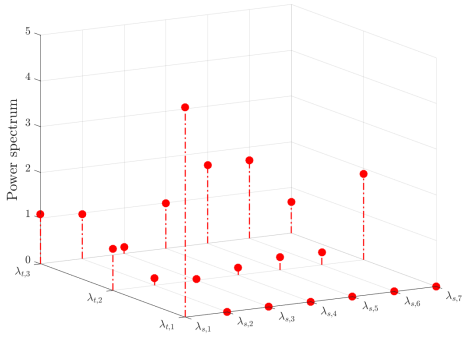
Next, we take the graph Laplacian as graph shift for G_S, G_T and $G_\times$. We now see on Figure 5.3 that the signal is also not bandlimited in the time frequency domain nor in the graph frequency domain. We see that the signal is 15-bandlimited in the joint frequency domain so again, the bound of N-bandlimitedness is not reached when using the graph Laplacian as graph shift.

Finally, taking the matrices $\boldsymbol{M}$, $\boldsymbol{M}_T$ and $\tilde{\boldsymbol{M}}$ as graph shifts of the graphs G_S, G_T and $G_\times$ respectively, we can observe on Figure 5.4 that the signal X is not bandlimited in the time frequency domain, it is 3-bandlimited in the graph frequency



(a) DFT

(b) GFT

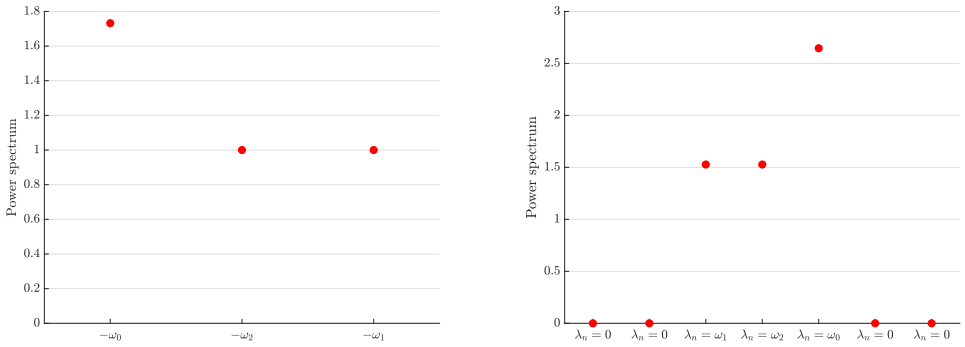


(c) JFT

Figure 5.3: Example 5.1.1 : Power spectrum of X obtained by the DFT, GFT and JFT when using the graph Laplacian as graph shift

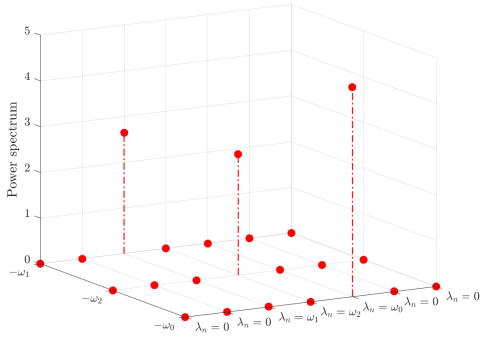
domain and it is also 3-bandlimited in the joint frequency domain. We see that the maximum bound of N stated in theorem 5.1.1 is verified and more specifically, the bound of $\dim(\mathcal{I}) = 3$ is attained. Indeed, we have $T = 3$ and we have three eigenvalues of $\boldsymbol{M}$ that are third root of unity. We also see that the values where the spectrum is nonzero on Figure 5.4c are situated on the diagonal where $\lambda_n - \omega_k = 0$, illustrating the resonance between time and space. △

Up until now, we have used either the adjacency matrix or the Laplacian matrix of the Cartesian product graph as the graph shift. We have seen in this section that there are other matrices that are suitable as graph shifts. In the case of linear periodic dynamical systems, using the periodic dynamical shift $\tilde{\boldsymbol{M}}$ provides results that are consistent with the intuition. Indeed, knowing the matrix $\boldsymbol{M}$, one could always sample the first N nodes (the initial condition x_0) and from these values, recover the signal over all time steps. The matrix $\boldsymbol{M}$ can be seen as another kind of adjacency matrix for the spatial graph where we only keep the edges and corresponding weights that are involved in the evolution of the signal. Matrix $\boldsymbol{M}_T$ which is a circulant matrix provides the notion of cyclic time hence of periodicity. With



(a) DFT

(b) GFT



(c) JFT

Figure 5.4: Example 5.1.1 : Power spectrum of X obtained by the DFT, GFT and JFT when using $\boldsymbol{M}_T$, $\boldsymbol{M}$ and $\tilde{\boldsymbol{M}}$ as graph shifts

these graph shifts, we can still exploit the structure of the product graph which reduces the computational costs since we only compute the eigendecompositions of $\boldsymbol{M} \in \mathbb{C}^{N\times N}$ and $\boldsymbol{M}_T \in \mathbb{C}^{T\times T}$ (and not the eigendecomposition of $\tilde{\boldsymbol{M}} \in \mathbb{C}^{NT\times NT}$) and we reduce the sufficient number of nodes to sample to $\dim(\mathcal{I}) \leq N$ with $\mathcal{I}$ the vector space generated by the eigenvectors of $\boldsymbol{M}$ corresponding to eigenvalues that are a T^{th} root of unity (this bound could not always be achieved with the conventional graph shifts such as the adjacency matrix or the graph Laplacian).

5.2 A characterization of valid sampling sets

In this section, we will study the sampling strategy for signals issued from dynamical systems with periodic constraints. From the previous section, we know that it is sufficient to sample N nodes from $G_\times$ to get perfect recovery of the whole signal X since periodic signals are at most N-bandlimited when using $\tilde{\boldsymbol{M}}$ from (5.3) as graph shift (theorems 5.1.1 and 5.1.3). Of course, the choice of the sampling set stays important since not all sets of N nodes will allow to get perfect recovery. The sampling

operator $\boldsymbol{\Psi}$ still has to verify the rank theorem 1.2.1, i.e. $rank(\boldsymbol{\Psi}\boldsymbol{V}_{(N)}) = N$ with $\boldsymbol{V}$ the eigenvector matrix of the graph shift.

One first option that is already available to us is to apply one of the two algorithms presented in section 2.1. These algorithms also work on such signals. The sampling set selection is then based on the singular values of the eigenvector matrix of $\tilde{\boldsymbol{M}}$ for the first algorithm and on theorem 1.2.1 for the second algorithm.

Since we are dealing with signals derived from dynamical systems and defined by a set of linear equations, we feel that we could understand better which nodes we should sample to get perfect recovery based on the dynamic of the system. Indeed, the first idea that comes to the mind in the case of these signals is to sample the N first nodes, i.e. the initial condition of the system. Knowing the matrix $\boldsymbol{M}$, we can then easily recover the whole signal by applying $\boldsymbol{M}$ repeatedly on the known initial conditions. But what if we cannot sample all the nodes at the first time step? In this section, we will show that we can gain some insights on the sampling set we should select by looking at the structure of a particular graph derived from the dynamical matrix $\boldsymbol{M}$. We will state a conjecture about a sufficient condition for a sampling set to enable perfect recovery of the whole signal X.

We still consider a periodic time-varying graph signal $X \in \mathbb{C}^{N\times T}$ whose columns satisfy (5.1) and (5.2). Let us define the network $G_D = (V_D, E_D)$ that indicates the dynamic of the signal. The adjacency matrix of this network is defined by

$$\boldsymbol{A}_{adj} = \begin{bmatrix} 0 & 0 & \dots & 0 & \boldsymbol{M} \\ \boldsymbol{M} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{M} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{M} & 0 \end{bmatrix}. \tag{5.7}$$

A nonzero value at the $(i,j)^{th}$ entry indicates that node i dynamically depends on node j at one time step distance. We thus obtain a directed dynamic graph. To illustrate this, we consider the following example.

Example 5.2.1. Let us consider the spatiotemporal network $G_\times = G_T \times G_S$ from example 5.1.1 where G_S is the seven-node undirected graph from Figure 5.1 and G_T is a ring graph with $T = 3$ nodes. We consider a time-varying graph signal X on this graph following the same dynamics as in the previous example, i.e. such that

$x_{t+1} = \boldsymbol{M} x_t$ with

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

and periodic constraint $x_0 = \boldsymbol{M} x_{T-1}$. The dynamical network G_D with adjacency matrix as in equation 5.7 is given on Figure 5.5. The nodes can be numbered from left to right and from above to below. To avoid crossing arrows, we represented the nodes at time 0 (from 1 to 7) above and below but let us note that because of the periodic constraint, these nodes should be represented only once and the dynamical graph G_D should be cyclic. In this figure, each node is linked to the nodes at the previous time step from which it derives its value. △

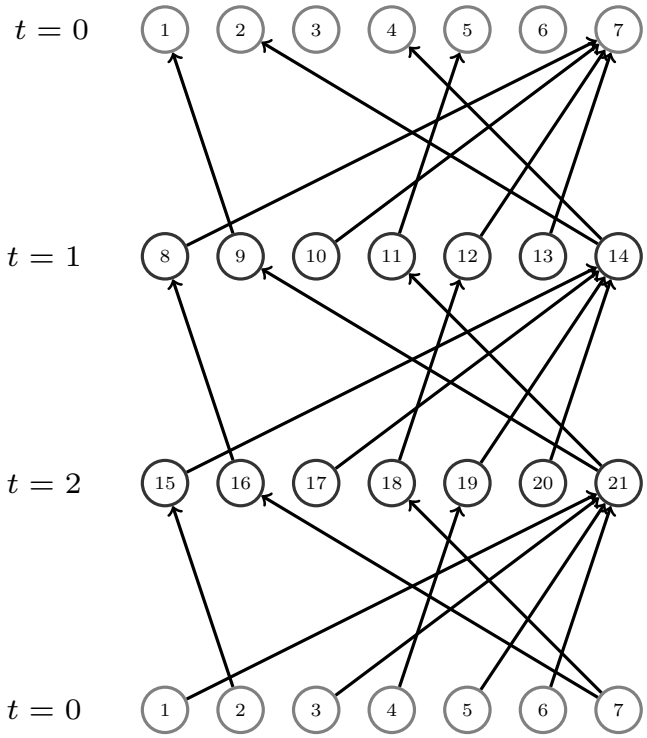


Figure 5.5: Dynamical periodic network of example 5.2.1

The dynamical network G_D can be understood as a representation of the linear equations deriving from the dynamical system. Each node is a variable and we have one linear equation linking each node to its out-going neighbors. In the case of a periodic signal, we have a system with NT variables and NT equations but this

system is under-determined. Its rank is lower than NT (it is $NT - \dim(\mathcal{I})$) thus we need to find the values of at most $\dim(\mathcal{I})$ variables in order to recover the values of all the other variables.

Intuitively, we see that at a certain time step and knowing the matrix $\boldsymbol{M}$, if all the signal values on the nodes from which a certain node i depends are known, then the value of this node can be recovered. Also, if the value of one node is known and all the values of the nodes from which it depends but one are also known, we can recover the value of this last node. These intuitive rules can be represented with some coloring rules on the dynamical graph G_D.

Suppose we have some nodes of G_D that are colored in red and other nodes are left white. The red nodes are those for which we already know the value (because of sampling) or for which we know that we can recover the value from other known nodes thanks to the dynamic. The white nodes are those for which we do not know the value and we do not know yet if we can recover the value from the other nodes. The two intuitions above can be formulated as follows :

Rule 5.2.1. If a node i is white and all its out-going neighbors are red, then i can be colored in red. This is illustrated on Figure 5.6.

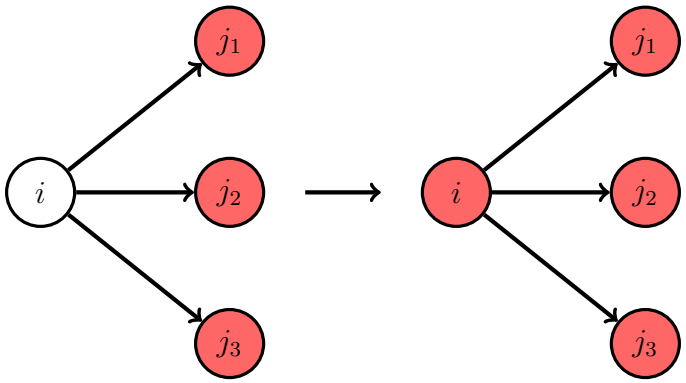


Figure 5.6: First rule

Rule 5.2.2. If a node i is red and if node j is its only out-going neighbor that is white, then j can be colored in red. This is illustrated on Figure 5.7.

With these two coloring rules, the problem of finding a sampling set sufficient to get perfect recovery of the dynamical signal X on the whole graph is equivalent to the problem of finding a set of nodes that, when colored in red, allows to color the whole graph G_D by using the rules.
The zero-forcing set problem is a similar problem. In this problem, only one color-change rule is commonly used. On a graph where each vertex is colored either white

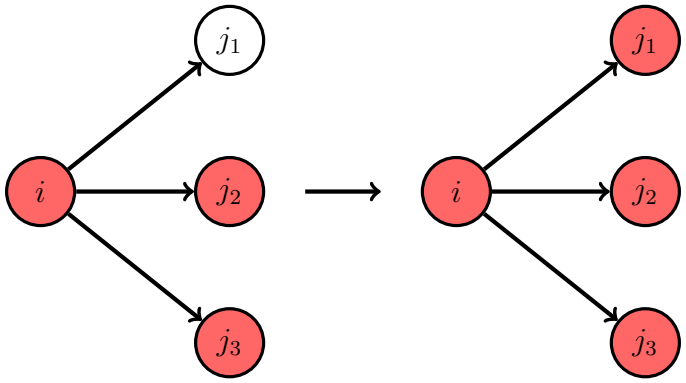


Figure 5.7: Second rule

or red, if i is a red vertex, and exactly one neighbor j of i is white, then change the color of j to red. The problem of finding a zero-forcing set on a simple graph G and more precisely the problem of computing $Z(G)$, the minimum size of a zero-forcing set of nodes on G is commonly used to bound the minimum rank of G. Let us associate to G all symmetric real matrices whose $(i,j)^{th}$ entry (for $i \neq j$) is nonzero whenever $\{i,j\}$ is an edge in G and is zero otherwise. Then the minimum rank of G is defined as the smallest possible rank over all these matrices [AIM Minimum Rank – Special Graphs Work Group, 2008]. We can see that our problem is similar to the problem of finding a zero-forcing set on the graph G_D except that we consider a graph that is directed, the matrices associated to this graph are not necessarily symmetric. The second rule 5.2.2 we use is an adaptation of the rule commonly used in the zero-forcing set problem for directed graphs and is already used to solve the zero-forcing set problem on directed graphs [Berliner et al., 2017]. However, our first rule 5.2.1 is an addition we made because it fits well to solve our problem but it does not seem to be a commonly used rule in the literature.

Let us apply these rules with an initial set on a simple example.

Example 5.2.2. Let us consider the spatiotemporal network $G_\times = G_T \times G_S$ where G_S is an undirected ring graph with 3 nodes and G_T is also a ring graph with $T = 3$ nodes. We consider a time-varying graph signal X on this graph such that $x_{t+1} = \boldsymbol{M} x_t$ with

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and periodic constraint $x_0 = \boldsymbol{M} x_{T-1}$. In this case, any initial condition x_0 yields a valid periodic signal X. From theorem 5.1.1, we know that we have to sample at most three nodes to recover the whole signal. The chosen sampling set is colored in red on the dynamical network shown on Figure 5.8. Let us recall that the upper level of nodes and the lower level are the same so there are indeed three different

nodes that are colored.

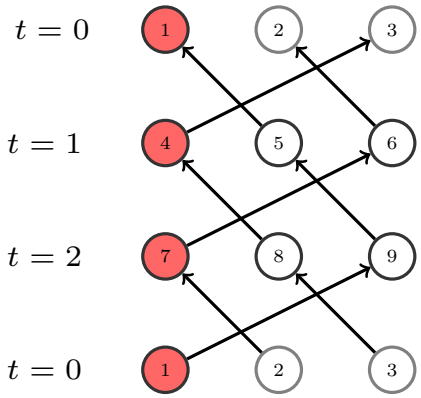


Figure 5.8: Dynamical periodic network and sampling set (in red) of example 5.2.2

Denoting by $(j_1, ..., j_k) \Rightarrow i$ when node i is colored by its out-going neighbors $(j_1, ..., j_k)$ following the first rule and by $i \rightarrow j$ when node j is colored by node i following the second rule, let us list the successive coloring steps that take place on this dynamical graph with the initial sampling set as shown.

We have :

- $1 \rightarrow 9$, $7 \rightarrow 6$, $4 \rightarrow 3$
- $1 \Rightarrow 5$, $4 \Rightarrow 8$, $7 \Rightarrow 2$

△

The two rules that we have introduced so far are understandable in terms of linear equations. Both rules can be formulated as follows : if there is only one unknown in a (linear) equation, the value of this unknown can be recovered from the values of the other variables present in this equation. When dealing with systems of linear equations, one often resorts to substitution, i.e. formulating one unknown in terms of other unknowns. On the dynamical graph G_D, we can represent substitution from node to node by coloring the node from which we substitute in one color and circling the node that we substitute in the same color.

Example 5.2.3. Suppose we have the following relations between nodes as shown on Figure 5.9 (left). Since the only unknown value to get the value of i is j_2, we can substitute the value of i by some value depending only on the unknown j_2. △

Let us note that it is not always obvious to recognize a full coloring set when given. However, knowing how the nodes dynamically interact with each other, it is sometimes easy for such toy examples as the ones used in this thesis to find a

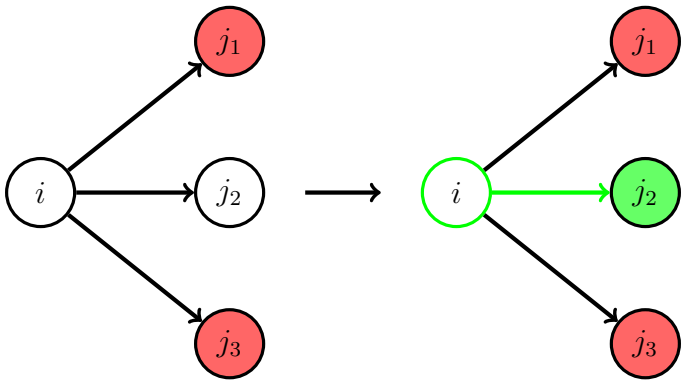


Figure 5.9: Substitution

sufficient sampling set. The purpose of this section was to characterize the sampling set of a signal issued from a dynamical system.

In the examples 5.2.1 and 5.2.2, we only used the matrix $\boldsymbol{M}$. In the first example, this matrix is not invertible. However, in general, it could be. If it is, we can also recover the signal by working backward in time. In general, we will have a sparse matrix $\boldsymbol{M}$ but $\boldsymbol{M}^{-1}$ will be full. We then get two sets of directed edges in our dynamical network : one set of edges corresponding to the matrix $\boldsymbol{M}$ in one way and a second set corresponding to the matrix $\boldsymbol{M}^{-1}$ in the other way. The adjacency matrix of the dynamical network then becomes

$$\boldsymbol{A}_{adj} = \begin{bmatrix} 0 & \boldsymbol{M}^{-1} & 0 & \dots & 0 & 0 & \boldsymbol{M} \\ \boldsymbol{M} & 0 & \boldsymbol{M}^{-1} & \dots & 0 & 0 & 0 \\ 0 & \boldsymbol{M} & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \boldsymbol{M}^{-1} & 0 \\ 0 & 0 & 0 & \dots & \boldsymbol{M} & 0 & \boldsymbol{M}^{-1} \\ \boldsymbol{M}^{-1} & 0 & 0 & \dots & 0 & \boldsymbol{M} & 0 \end{bmatrix}. \tag{5.8}$$

We still have that a nonzero value at the $(i,j)^{th}$ entry indicates that node i dynamically depends on node j at one time step distance but now we also look at time going backward. The rules developed above can still be applied in the same way on this double dynamical network.

Finally, let us note that in the coloring problem, we are only interested in the structure of the adjacency matrix $\boldsymbol{A}_{adj}$ (from (5.7) or (5.8) if $\boldsymbol{M}$ is invertible) of G_D, i.e. only on the place of its nonzero entries and not on their values. The edges of G_D are unweighted. A sampling set that is sufficient for a signal following a certain dynamical structure will also be sufficient for another signal that follows the same dynamical structure but weighted differently or with other initial conditions. We

now state a conjecture about the characterization of a valid (sufficient) sampling set.

Conjecture 5.2.1. *Given a periodic time-varying graph signal $X \in \mathbb{C}^{N\times T}$ whose columns satisfy*

$$x_{t+1} = \boldsymbol{M}x_t \in \mathbb{C}^N$$

and

$$x_0 = \boldsymbol{M}x_{T-1}$$

with $t \in \{0, ..., T-2\}$ and x_t the t^{th} column of X, we denote by $G_\times$ the spatiotemporal graph on which the signal X resides and by G_D the dynamical graph of X with adjacency matrix given by (5.7) or by (5.8) if $\boldsymbol{M}$ is invertible. A sampling set on $G_\times$ allows to get perfect recovery of the whole signal X when using $\tilde{\boldsymbol{M}}$ from (5.3) as graph shift for $G_\times$ if and only if this sampling set is a full coloring set on G_D.

5.3 Stability analysis

In this section, we study the stability of the linear dynamical system from which the time-varying graph signal X is issued and how it affects the sampling strategy to apply.

When having the following linear discrete-time dynamical system

$$x_{t+1} = \boldsymbol{M}x_t,$$

we say that this system is exponentially stable if all its solutions tend exponentially to zero and it is stable if all its solutions are bounded. It is unstable if there exist unbounded solutions and if there exist unbounded solutions that tend exponentially to infinity, we say that the instability is exponential [Halanay and Rasvan, 2000]. Since we consider time-varying graph signals X issued from periodic dynamical systems of the form $x_{t+1} = \boldsymbol{M}x_t$, $t = 0, ..., T-2$, with $x_0 = \boldsymbol{M}x_{T-1}$, we face the case where the linear systems from which the periodic signals are issued are always stable in the sense that all the solutions of the systems are bounded.

In section 5.1, we saw that using the matrix $\tilde{\boldsymbol{M}}$ from equation (5.3) as graph shift resulted in a bandwidth of at most $d = \dim(\mathcal{I})$ in the joint Fourier domain for the signal X with $\mathcal{I}$ the vector space generated by the eigenvectors v_n of $\boldsymbol{M}$ corresponding to the eigenvalues λ_n such that λ_n is a T^{th} root of unity (theorem 5.1.3). Since $\text{vec}(X) = x \in \text{null}(\tilde{\boldsymbol{M}})$, the zero frequency components of x correspond to the zero eigenvalues of $\tilde{\boldsymbol{M}}$. A valid sampling set should verify the rank theorem 1.2.1 from section 1.2. There are often many different possible sets of nodes that satisfy this condition. Moreover in section 2.1, we have seen that an optimal sampling operator (optimal in the sense that it minimizes the effect of noise) should select the nodes

in order to maximize the minimum singular value of the reduced eigenvector matrix of the graph shift, i.e.

$$\boldsymbol{\Psi}^{opt} = \arg\max_{\boldsymbol{\Psi}} \sigma_{\min}(\boldsymbol{\Psi}\boldsymbol{V}_{(K)})$$

We can thus sample $d = \dim(\mathcal{I})$ nodes based only on the eigenvectors of $\tilde{\boldsymbol{M}}$ corresponding to the zero eigenvalues. From the characterization of the eigendecomposition of $\tilde{\boldsymbol{M}}$ in section 5.1, we have seen that its eigenvectors can be defined by block and that these blocks are multiple of each other. Indeed, we have

$$\tilde{v}_{n,k} = \begin{bmatrix} v_n & \omega_k v_n & \omega_k^2 v_n & \dots & \omega_k^{T-1} v_n \end{bmatrix}$$

with v_n the eigenvectors of $\boldsymbol{M}$. Hence, the reduced eigenvector matrix (where we only select the d eigenvectors corresponding to the d zero eigenvalues, i.e., when $\lambda_n = \omega_k$) of the graph shift $\tilde{\boldsymbol{M}}$ can be written as

$$\tilde{\boldsymbol{V}}_{(d)} = \begin{bmatrix} v_1 & v_2 & \dots & v_d \\ \lambda_1 v_1 & \lambda_2 v_2 & \dots & \lambda_d v_d \\ \lambda_1^2 v_1 & \lambda_2^2 v_2 & \dots & \lambda_d^2 v_d \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{T-1} v_1 & \lambda_2^{T-1} v_2 & \dots & \lambda_d^{T-1} v_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{(d)} \\ \boldsymbol{V}_{(d)}\boldsymbol{\Lambda}_{(d)} \\ \boldsymbol{V}_{(d)}\boldsymbol{\Lambda}_{(d)}^2 \\ \vdots \\ \boldsymbol{V}_{(d)}\boldsymbol{\Lambda}_{(d)}^{T-1} \end{bmatrix}$$

Since we only look at the eigenvectors v_n, $n \in \{1, ..., d\}$, corresponding to the eigenvalues λ_n such that λ_n is a T^{th} root of unity, we have that $|\lambda_n| = 1$, $n \in \{1, ..., d\}$. Each block of $\tilde{\boldsymbol{V}}_{(d)}$ has entries with the same magnitudes as the other blocks. If we want to sample d nodes (select d rows in $\tilde{\boldsymbol{V}}_{(d)}$) in order to maximize the minimum singular value of $\tilde{\boldsymbol{V}}_{(d)}$, we can thus deduce that it does not impact the optimality of the sampling set to sample nodes at a certain time instant or at another as long as the sampling operator satisfies the rank theorem 1.2.1, i.e. as long as the sampling set is valid. Indeed, when facing a periodic signal, there is no real beginning and end to the signal so the time instant does not matter.

5.4 Summary

Given a periodic time-varying graph signal $X \in \mathbb{C}^{N \times T}$ whose columns satisfy

$$x_{t+1} = \boldsymbol{M} x_t \in \mathbb{C}^N,$$

with $t \in \{0, ..., T-2\}$ and

$$x_0 = \boldsymbol{M} x_{T-1}$$

and given $\mathcal{I}$ the vector space generated by the eigenvectors v_n of $\boldsymbol{M}$ corresponding to the eigenvalues λ_n such that λ_n is a T^{th} root of unity, we showed in section 5.1 that X is at most d-bandlimited when using $\tilde{\boldsymbol{M}}$ as graph shift, with $d = \dim(\mathcal{I})$.

A valid sampling operator $\boldsymbol{\Psi}$ should select nodes such that $rank(\boldsymbol{\Psi}\tilde{\boldsymbol{V}}_{(d)}) = d$ with $\tilde{\boldsymbol{V}}$ the eigenvector matrix of $\tilde{\boldsymbol{M}}$. There are often multiple sampling sets possible that satisfy this condition. We showed in section 5.2 how some coloring rules on the dynamical graph G_D issued from $\boldsymbol{M}$ could give us some insights about the validity of a particular sampling set.
Finally, an optimal sampling operator minimizing the effect of noise should maximize the minimum singular value of $\tilde{\boldsymbol{V}}_{(d)}$. We explained by an argument about stability why it was not more optimal to sample nodes from a certain time step compared to nodes from another time step in section 5.3.

Chapter 6

Signals issued from linear dynamical systems without periodic constraints

Now that we have studied in chapter 5 how we can efficiently sample periodic signals issued from linear dynamical systems, we focus in this chapter on the case where there is no periodic constraint. The signals we study are non-periodic signals issued from linear dynamical systems.

6.1 Sampling and reconstruction

In this section, we study the problem of sampling and reconstruction of a signal that is issued from a linear dynamical system without any periodic constraint (contrary to section 5.1).

We consider the Cartesian product graph $G_\times = G_T \times G_S$ between our temporal graph $G_T = (V_T, E_T)$ and our spatial graph $G_S = (V_S, E_S)$. The time-varying graph signals $X \in \mathbb{C}^{N \times T}$ that we now consider are signals derived from linear dynamical systems, i.e. the columns of X satisfy

$$x_{t+1} = \boldsymbol{M} x_t \in \mathbb{C}^N \tag{6.1}$$

with $t \in \{0, ..., T-2\}$ and x_t the t^{th} column of X but we do not add any periodic constraint. Let us recall that the columns of X are the graph signals at different time instants and the rows of X are time signals on different nodes. The temporal graph G_T in this section will always be a path graph with T nodes. Let us note now that every initial condition $x_0 \in \mathbb{C}^N$ is valid since we do not have any constraint of periodicity. In the following, we note $\text{vec}(X) = x$.

Similarly to what we did in the periodic case of section 5.1, we will define a new

graph shift which guarantees that we have a sparse representation in the frequency domain of a signal X issued from a linear dynamical system. Let us note that taking $\tilde{\boldsymbol{M}}$ from (5.3), we do not have $\tilde{\boldsymbol{M}}x = 0$ anymore because we do not have any periodic constraint in this case. We thus have to derive another matrix.

Theorem 6.1.1. *Given a non-periodic time-varying graph signal $X \in \mathbb{C}^{N\times T}$ whose columns satisfy* (6.1)*, X is N-bandlimited at most in the joint Fourier domain of $G_\times$ when using*

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M}^*\boldsymbol{M} & -\boldsymbol{M}^* & 0 & \dots & 0 & 0 & 0 \\ -\boldsymbol{M} & \boldsymbol{I}+\boldsymbol{M}^*\boldsymbol{M} & -\boldsymbol{M}^* & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\boldsymbol{M} & \boldsymbol{I}+\boldsymbol{M}^*\boldsymbol{M} & -\boldsymbol{M}^* \\ 0 & 0 & 0 & \dots & 0 & -\boldsymbol{M} & \boldsymbol{I} \end{bmatrix} \tag{6.2}$$

as graph shift. $\bar{\boldsymbol{M}}$ will be referred to as the non-periodic dynamical graph shift.

Proof. Let us define

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{M} & -\boldsymbol{I} & \dots & 0 & 0 \\ 0 & 0 & \boldsymbol{M} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \boldsymbol{M} & -\boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{N(T-1)\times NT} \tag{6.3}$$

This matrix is such that

$$\boldsymbol{\Pi}x = 0$$

hence $x \in \text{null}(\boldsymbol{\Pi})$. We can verify that all rows of $\boldsymbol{\Pi}$ are linearly independent hence $rank(\boldsymbol{\Pi}) = N(T-1)$ hence from the rank-nullity theorem, we have that dim null$(\boldsymbol{\Pi}) = N$. Since $\boldsymbol{\Pi}$ is not a square matrix, we cannot take this matrix as graph shift. However, we note that x is bandlimited with a bandwidth of at most N when taking the right singular vector matrix $\boldsymbol{V}$ from $\boldsymbol{\Pi} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ as Fourier basis. Hence, we define

$$\bar{\boldsymbol{M}} = \boldsymbol{\Pi}^*\boldsymbol{\Pi} = \boldsymbol{V}(\boldsymbol{\Sigma}^*\boldsymbol{\Sigma})\boldsymbol{V}^* \tag{6.4}$$

The matrix $\bar{\boldsymbol{M}} \in \mathbb{R}^{NT\times NT}$ also satisfy $x \in \text{null}(\bar{\boldsymbol{M}})$ and dim null$(\bar{\boldsymbol{M}}) = N$. From this, we can directly deduce that x will be bandlimited with a bandwidth of at most N when taking $\bar{\boldsymbol{M}}$ as the graph shift for $G_\times$. □

To conclude this section, we see that just like in the periodic case, there exist a graph shift matrix in the non-periodic case that allows to get a sparse representation of the time-varying graph signal X in the joint Fourier domain. This matrix is $\bar{\boldsymbol{M}}$ which is built from the dynamical matrix $\boldsymbol{M}$. When using this matrix as graph shift, the sufficient number of nodes to sample in order to get perfect recovery of

the whole signal is N. However, we cannot exploit the structure of the product graph when using this matrix as graph shift because it cannot be expressed as a Kronecker sum (in contrary to the graph shift $\tilde{\boldsymbol{M}}$ in the case of periodic signals) so we cannot sample in time and space separately when selecting the sampling set and the computational cost increases.

6.2 A characterization of valid sampling sets

In this section, we will study the sampling strategy for signals issued from dynamical systems without periodic constraints. From the previous section, we know that it is sufficient to sample N nodes in the case of non-periodic signals to get perfect recovery since the non-periodic signals are at most N-bandlimited when using $\bar{\boldsymbol{M}}$ from (6.2) as graph shift. Of course, the choice of the sampling set stays important since not all sets of N nodes will allow to get perfect recovery. The sampling operator $\boldsymbol{\Psi}$ still has to verify the rank theorem 1.2.1, i.e. $rank(\boldsymbol{\Psi}\boldsymbol{V}_{(N)}) = N$ with $\boldsymbol{V}$ the eigenvector matrix of the graph shift.

The notions presented in this section are very similar to the ones presented in section 5.2. As in the periodic case, a first option that is already available to us is to apply one of the two algorithms presented in section 2.1. But again, since we are dealing with signals derived from dynamical systems and defined by a set of linear equations, we feel that we could understand better which nodes we should sample to get perfect recovery based on the dynamic of the system.

We still consider a time-varying graph signal X with columns satisfying $x_{t+1} = \boldsymbol{M}x_t$. For such signals, we now define the network $G_D = (V_D, E_D)$ that indicates the dynamic of the signal as the network with adjacency matrix defined by

$$\boldsymbol{A}_{adj} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ \boldsymbol{M} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{M} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{M} & 0 \end{bmatrix} \tag{6.5}$$

if $\boldsymbol{M}$ is non invertible and by

$$\boldsymbol{A}_{adj} = \begin{bmatrix} 0 & \boldsymbol{M}^{-1} & 0 & \dots & 0 & 0 & 0 \\ \boldsymbol{M} & 0 & \boldsymbol{M}^{-1} & \dots & 0 & 0 & 0 \\ 0 & \boldsymbol{M} & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \boldsymbol{M}^{-1} & 0 \\ 0 & 0 & 0 & \dots & \boldsymbol{M} & 0 & \boldsymbol{M}^{-1} \\ 0 & 0 & 0 & \dots & 0 & \boldsymbol{M} & 0 \end{bmatrix} \tag{6.6}$$

if $\boldsymbol{M}$ is invertible. Again, a nonzero value at the $(i,j)^{th}$ entry indicates that node i dynamically depends on node j at one step distance (forward or backward in time). We thus obtain a directed dynamical graph. To illustrate this, we consider the following example.

Example 6.2.1. Let us consider the spatiotemporal network $G_\times$ presented on Figure 6.1. The spatial network is a simple undirected path with 5 nodes. Time is also represented by a path with 4 nodes. The nodes can be numbered from left to right and from above to below. For simplicity, we do not represent any signal on this graph but any $x \in \mathbb{C}^5$ can be used as initial condition x_0 on the nodes v_1 to v_5. The signal on this graph evolves following a non-periodic dynamical system of the form $x_{t+1} = \boldsymbol{M}x_t$ with

$$\boldsymbol{M} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

This is a simple example of a consensus process where each node updates its value according to a weighted sum of the values of its neighbors and itself. The dynamical network G_D with adjacency matrix as in equation 6.5 is given on Figure 6.2. In this figure, each node is linked to the nodes at the previous time step from which it derives its value. △

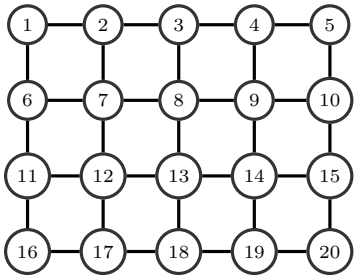


Figure 6.1: Spatiotemporal network of example 6.2.1

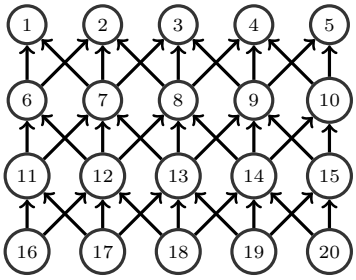


Figure 6.2: Dynamical network of example 6.2.1

The dynamical network G_D can be understood as a representation of the linear equations deriving from the dynamical system. Each node is a variable and we have one linear equation linking each node to its out-going neighbors. In the case of a non-periodic signal, we hence face a system with NT variables and $N(T-1)$ linear

equations. Using the adjacency matrix, we have

$$\boldsymbol{A}x = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ \boldsymbol{M} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{M} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{M} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{T-1} \end{bmatrix} = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ \vdots \\ x_{T-1} \end{bmatrix}$$

We thus have to get the values of N variables in order to recover the values of all the other variables. These variables cannot be arbitrarily chosen, they have to allow us to recover the other variables.

The same two rules presented in section 5.2 can be used in the case of a non-periodic signal. Let us recall them. Suppose we have some nodes of G_D that are colored in red and other nodes are left white. The red nodes are those for which we already know the value (because of sampling) or for which we know that we can recover the value from other known nodes thanks to the dynamic. The white nodes are those for which we do not know the value and we do not know yet if we can recover the value from the other nodes. The two rules are :

Rule 6.2.1. If a node i is white and all its out-going neighbors are red, then i can be colored in red.

Rule 6.2.2. If a node i is red and if node j is its only out-going neighbor that is white, then j can be colored in red.

Let us apply these rules with an initial set on the same example as above.

Example 6.2.2. We consider the same spatiotemporal network and the same time-varying graph signal as in example 6.2.1. Since we are looking at a non-periodic signal, we know that we have to sample at most N nodes to recover the whole signal. The chosen sampling set is colored in red on the dynamical network shown on Figure 6.3.

Denoting by $(j_1, ..., j_k) \Rightarrow i$ when node i is colored by its out-going neighbors $(j_1, ..., j_k)$ following the first rule and by $i \rightarrow j$ when node j is colored by node i following the second rule, let us list the successive coloring steps that take place on this dynamical graph with the initial sampling set as shown.

We have :

- $16 \rightarrow 12$, $11 \rightarrow 7$, $6 \rightarrow 2$
- $12 \rightarrow 8$, $7 \rightarrow 3$
- $8 \rightarrow 4$

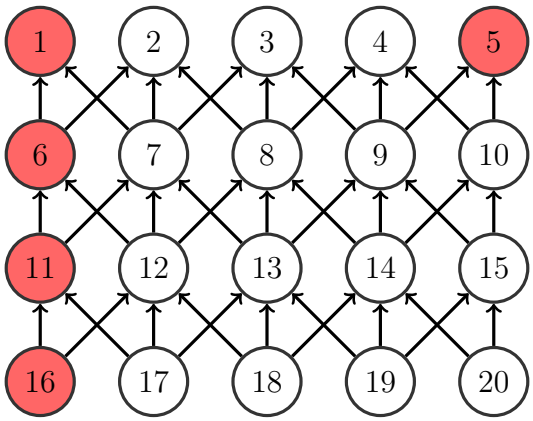


Figure 6.3: Dynamical network and sampling set (in red) of example 6.2.2

- $(4,5) \Rightarrow 10$, $(3,4,5) \Rightarrow 9$
- $(9,10) \Rightarrow 15$, $(8,9,10) \Rightarrow 14$, $(7,8,9) \Rightarrow 13$
- $(14,15) \Rightarrow 20$, $(13,14,15) \Rightarrow 19$, $(12,13,14) \Rightarrow 18$, $(11,12,13) \Rightarrow 17$

△

Let us also recall how we can represent the process of substitution on the dynamical graph G_D : we can represent substitution from node to node by coloring the node from which we substitute in one color and circling the node that we substitute in the same color.

Example 6.2.3. Suppose we have the following relations between nodes as shown on Figure 6.4 (left). Since the only unknown value to get the value of i is j_2, we can substitute the value of i by some value depending only on the unknown j_2. △

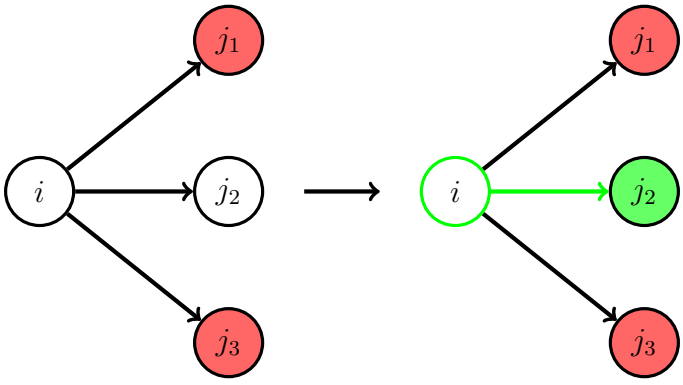


Figure 6.4: Substitution

Let us apply this principle of substitution, again to the same example as before but with another sampling set.

Example 6.2.4. We now look at another sampling set (in red on Figure 6.5). Using substitutions, we can propagate the unknown values of nodes 2 and 4 to the known values of nodes 16 and 18. Hence, we can recover the two unknown values from these known values since it is the same as having two equations of two unknowns. When having recovered the values of nodes 2 and 4, we can then recover the whole signal using rule 5.2.2.

△

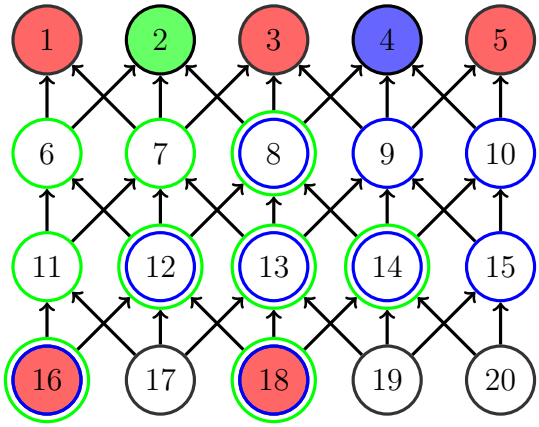


Figure 6.5: Dynamical network and substitution process of example 6.2.4

Again, let us emphasize the fact that in the examples 6.2.1, 6.2.2 and 6.2.4, the matrix $\boldsymbol{M}$ that we used was not invertible but that, in general, it could be so we could also work backward in time. We now state a similar conjecture as the one presented in the periodic case (Conjecture 5.2.1) about the characterization of a valid (sufficient) sampling set for a non-periodic signal.

Conjecture 6.2.1. *Given a non-periodic time-varying graph signal* $X \in \mathbb{C}^{N\times T}$ *whose columns satisfy*

$$x_{t+1} = \boldsymbol{M}x_t \in \mathbb{C}^N$$

with $t \in \{0, ..., T-2\}$ *and* x_t *the* t^{th} *column of* X*, we denote by* $G_\times$ *the spatiotemporal graph on which the signal* X *resides and by* G_D *the dynamical graph of* X *with adjacency matrix given by* (6.5) *or by* (6.6) *if* $\boldsymbol{M}$ *is invertible. A sampling set on* $G_\times$ *allows to get perfect recovery of the whole signal* X *when using* $\bar{\boldsymbol{M}}$ *from* (6.2) *as graph shift for* $G_\times$ *if and only if this sampling set is a full coloring set on* G_D*.*

6.3 Stability analysis

In this section, we study the stability of the dynamical systems and how it affects the sampling strategy to apply. We show that it is reasonable to assume that for a stable dynamical system, it is better to sample the first N nodes when possible while for unstable systems, we should also try to sample nodes at the end of the signal. First, we recall some notions about the stability of a linear discrete-time dynamical system [Halanay and Rasvan, 2000].

Let us consider the following linear discrete-time dynamical system

$$x_{t+1} = \boldsymbol{M} x_t.$$

We say that this system is exponentially stable if all its solutions tend exponentially to zero and it is stable if all its solutions are bounded. It is unstable if there exist unbounded solutions and if there exist unbounded solutions that tend exponentially to infinity, we say that the instability is exponential [Halanay and Rasvan, 2000].

We have the following theorem concerning the stability of this system [Halanay and Rasvan, 2000].

Theorem 6.3.1. *The system is exponentially stable if and only if the eigenvalues of $\boldsymbol{M}$ are inside the unit disk of the complex plane, i.e., $|\lambda_i(\boldsymbol{M})| < 1, \forall i$ which can also be formulated as $\rho(\boldsymbol{M}) < 1$. The system is stable if and only if the eigenvalues of $\boldsymbol{M}$ are inside the unit disk and on the circle, those on the circle being located in one-dimensional Jordan cells. The system is unstable if at least one eigenvalue on the unit circle is located in a Jordan cell of dimension larger than 1. The instability is exponential if at least one eigenvalue is located outside the unit disk, i.e., $\rho(\boldsymbol{M}) > 1$.*

Let us consider a time-varying graph signal X issued from a non-periodic dynamical system of the form $x_{t+1} = \boldsymbol{M} x_t$. In the case of a non-periodic system, we can take any arbitrary matrix to be the matrix $\boldsymbol{M}$ hence we have that $\boldsymbol{M}$ can be either stable or unstable. We will suppose in this section that $\boldsymbol{M}$ is invertible.

In section 6.1, we saw that using the matrix $\bar{\boldsymbol{M}}$ from equation (6.2) as graph shift resulted in a bandwidth of at most N in the joint Fourier domain for the signal X. Since $\text{vec}(X) = x \in \text{null}(\bar{\boldsymbol{M}})$, the zero frequency components of x correspond to the N zero eigenvalues of $\bar{\boldsymbol{M}}$. A valid sampling set should verify the rank theorem 1.2.1 from section 1.2. There are often many different possible sets of nodes that satisfy this condition. Moreover, we have seen in section 2.1 that an optimal sampling operator (optimal in the sense that it minimizes the effect of noise) should select the nodes in order to maximize the minimum singular value of the reduced eigenvector matrix of the graph shift, i.e.

$$\boldsymbol{\Psi}^{opt} = \arg\max_{\boldsymbol{\Psi}} \sigma_{\min}(\boldsymbol{\Psi} \boldsymbol{V}_{(K)}) \tag{6.7}$$

We can thus sample N nodes based only on the eigenvectors of $\bar{\boldsymbol{M}}$ corresponding to the N zero eigenvalues. In this section, we will develop the structure of these eigenvectors based on the particular structure of $\bar{\boldsymbol{M}}$ which is block tridiagonal.

To support our developments, we will need several results on the eigendecomposition of block tridiagonal matrices [Sandryhaila and Moura, 2013]. Suppose an $N \times N$ block tridiagonal matrix of the form

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{D}_0 & & \\ \boldsymbol{C}_0 & \boldsymbol{B}_1 & \ddots & \\ & \ddots & \ddots & \boldsymbol{D}_{L-2} \\ & & \boldsymbol{C}_{L-2} & \boldsymbol{B}_{L-1} \end{bmatrix} \tag{6.8}$$

with $\boldsymbol{B}_n, \boldsymbol{C}_n, \boldsymbol{D}_n \in \mathbb{C}^{K \times K}$. We define $L = N/K$ and impose that

$$\det(\boldsymbol{D}_n) \neq 0, \quad n \geq 0. \tag{6.9}$$

We then define the family of $K \times K$ matrix polynomials $\boldsymbol{P}_n(x)$ that satisfy the relation

$$x \cdot \boldsymbol{P}_n(x) = \boldsymbol{C}_{n-1}\boldsymbol{P}_{n-1}(x) + \boldsymbol{B}_n\boldsymbol{P}_n(x) + \boldsymbol{D}_n\boldsymbol{P}_{n+1}(x) \tag{6.10}$$

with initial conditions $\boldsymbol{P}_{-1}(x) = \boldsymbol{0}_K$ and $\boldsymbol{P}_0(x) = \boldsymbol{I}_K$. This relation can be rewritten as

$$\boldsymbol{P}_{n+1}(x) = \boldsymbol{D}_n^{-1}(x \cdot \boldsymbol{P}_n(x) - \boldsymbol{B}_n\boldsymbol{P}_n(x) - \boldsymbol{C}_{n-1}\boldsymbol{P}_{n-1}(x)) \tag{6.11}$$

From these, we have the following theorem that allows us to compute the eigendecomposition of $\boldsymbol{A}$ [Sandryhaila and Moura, 2013].

Theorem 6.3.2. *Consider an arbitrary block tridiagonal matrix $\boldsymbol{A}$ of the form* (6.8) *satisfying* (6.9) *and the corresponding matrix polynomials $\boldsymbol{P}_n(x)$ generated by the recurrence* (6.11)*. Then v is an eigenvector of $\boldsymbol{A}$ that corresponds to an eigenvalue λ if and only if λ is a zero of the matrix polynomial $\boldsymbol{P}_L(x)$ and the vector v has the form*

$$v = \begin{bmatrix} \boldsymbol{P}_0(\lambda) \\ \vdots \\ \boldsymbol{P}_{L-2}(\lambda) \\ \boldsymbol{P}_{L-1}(\lambda) \end{bmatrix} u, \tag{6.12}$$

where $u \in \mathbb{C}^K$ is a vector from the null-space of the scalar matrix $\boldsymbol{P}_L(\lambda)$, i.e., the vector u satisfies $\boldsymbol{P}_L(\lambda)u = 0$.

Let us now apply theorem 6.3.2 to our tridiagonal matrix $\bar{\boldsymbol{M}}$.

We consider the matrix $\bar{\boldsymbol{M}} \in \mathbb{R}^{NT\times NT}$ where each block is of size $N \times N$ hence we have that $L = NT/N = T$. We also have that

$$\begin{aligned}
\boldsymbol{B}_0 &= \boldsymbol{M}^*\boldsymbol{M}, \\
\boldsymbol{B}_1 &= ... = \boldsymbol{B}_{T-2} = \boldsymbol{I} + \boldsymbol{M}^*\boldsymbol{M}, \\
\boldsymbol{B}_{T-1} &= \boldsymbol{I}, \\
\boldsymbol{D}_n &= -\boldsymbol{M}^* \quad \forall n, \\
\boldsymbol{C}_n &= -\boldsymbol{M} \quad \forall n.
\end{aligned}$$

We impose that $\det(-\boldsymbol{M}^*) \neq 0$, i.e. $\boldsymbol{M}$ is invertible. We want to compute the eigenvectors of $\bar{\boldsymbol{M}}$ corresponding to the zero eigenvalues. From before, we know that there are N zero eigenvalues and hence '0' is a zero of the matrix polynomial $\boldsymbol{P}_L(x)$. We have to compute $\boldsymbol{P}_0(0), \boldsymbol{P}_1(0), ..., \boldsymbol{P}_{T-1}(0)$. Computing these recursively using (6.11), it can be shown that the recurrence can be simplified and written as

$$\boldsymbol{P}_n(0) = \boldsymbol{M}\boldsymbol{P}_{n-1}(0) = \boldsymbol{M}^n, \quad 0 \leq n \leq T-1$$

Thus, the eigenvectors of $\bar{\boldsymbol{M}}$ corresponding to the zero eigenvalues have the form

$$\bar{v} = \begin{bmatrix} \boldsymbol{I}_N \\ \boldsymbol{M} \\ \boldsymbol{M}^2 \\ \vdots \\ \boldsymbol{M}^{T-1} \end{bmatrix} u$$

with u satisfying $\boldsymbol{P}_T(0)u = 0$. We still have to compute $\boldsymbol{P}_T(0)$:

$$\begin{aligned}
\boldsymbol{P}_T(0) &= -(\boldsymbol{M}^*)^{-1}(-\boldsymbol{I}\boldsymbol{P}_{T-1}(0) + \boldsymbol{M}\boldsymbol{P}_{T-2}(0)) \\
&= \boldsymbol{M}^{*-1}\boldsymbol{M}^{T-1} - \boldsymbol{M}^{*-1}\boldsymbol{M}\boldsymbol{M}^{T-2} \\
&= \boldsymbol{0}_N
\end{aligned}$$

Hence, $\text{null}(\boldsymbol{P}_T(0)) = \mathbb{C}^N$ is of dimension N and we can take N linearly independent vectors $u_0, ..., u_{N-1}$ to obtain the N linearly independent eigenvectors corresponding to the N zero eigenvalues of $\bar{\boldsymbol{M}}$. A simple choice is to take $\boldsymbol{U} = \boldsymbol{I}_N$ and we finally obtain

$$\bar{\boldsymbol{V}}_{(N)} = \begin{bmatrix} \boldsymbol{I}_N \\ \boldsymbol{M} \\ \boldsymbol{M}^2 \\ \vdots \\ \boldsymbol{M}^{T-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}\boldsymbol{I}_N\boldsymbol{V}^{-1} \\ \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \\ \boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^{-1} \\ \vdots \\ \boldsymbol{V}\boldsymbol{\Lambda}^{T-1}\boldsymbol{V}^{-1} \end{bmatrix} \tag{6.13}$$

We base the following discussion on the fact that the eigenvalues and the singular values of a matrix have "the same size". We also consider the fact that adding a row

of a "small" matrix to an existing matrix adds a "small" singular value while adding a row of a "large" matrix adds a "large" singular value. The following comments are only qualitative and provide some rules of thumbs when sampling nodes on a graph based on a signal that is issued from a dynamical system with matrix $\boldsymbol{M}$. The following rules of thumbs do not give an optimal sampling set but it seems to give a good approximation of what we can obtain when applying Algorithm 1 (which has been shown to give a good approximation to the optimal sampling operator from equation 6.7 [Avron and Boutsidis, 2013]). We will first consider the case where $\boldsymbol{M}$ is stable and then the case where it is unstable. We will also illustrate our assumptions with some small numerical examples. We suppose that we sample exactly N nodes.

6.3.1 Stable case

If $\boldsymbol{M}$ is asymptotically stable, i.e., $\rho(\boldsymbol{M}) < 1$, we have that $\lim_{k\to\infty} \boldsymbol{M}^k = 0$. We know from section 1.2 and more precisely from the rank theorem 1.2.1 that our sampling operator $\boldsymbol{\Psi}$ should be such that $rank(\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)}) = N$. If our conjecture 6.2.1 from this section is correct, this would be equivalent to selecting a set of nodes that is a full coloring set for the dynamical graph G_D. Besides that, it is also reasonable to assume that we should choose the N first nodes in order to maximize the minimum singular value of $\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)}$. Indeed, looking at the second form in equation (6.13) of $\bar{\boldsymbol{V}}_{(N)}$ where we have written the eigendecomposition of $\boldsymbol{M}$, we see that the eigenvalues of the different blocks become increasingly smaller and so do the singular values. The same reasoning holds when $\boldsymbol{M}$ is stable.

Example 6.3.1. We generate 1000 different time-varying graph signals $X \in \mathbb{C}^{N\times T}$ issued from a non-periodic dynamical system of the form $x_{t+1} = \boldsymbol{M}x_t$ such that the system is stable or asymptotically stable (i.e. $\rho(\boldsymbol{M}) \leq 1$). The matrix $\boldsymbol{M}$ is generated randomly at each iteration such that it always verifies $\rho(\boldsymbol{M}) \leq 1$. We take $N = 10$ and $T = 20$. At each iteration, we sample 10 nodes using Algorithm 1. Figure 6.6 shows which nodes were sampled over all the iterations. We can see that 75% of all the nodes sampled (10 nodes at each of the 1000 iterations) were sampled at the first time instant. Figure 6.7 shows the relative reconstruction error committed at each iteration when sampling N nodes with Algorithm 1 and when sampling the N first nodes. We can see that the error committed when sampling the N first nodes is similar as the one committed with Algorithm 1 indicating that sampling the N first nodes when the signal is issued from a stable system gives a good approximation to the optimal sampling set. △

6.3.2 Unstable case

Now if $\boldsymbol{M}$ is asymptotically unstable, i.e., $\rho(\boldsymbol{M}) > 1$, $\lim_{k\to\infty} ||\boldsymbol{M}^k|| = \infty$. We can then have eigenvalues that are stable and others that are not. In that case,

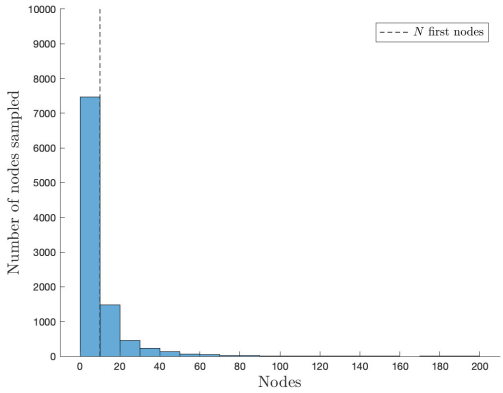


Figure 6.6: Histogram of the number of nodes sampled when the system is stable in example 6.3.1

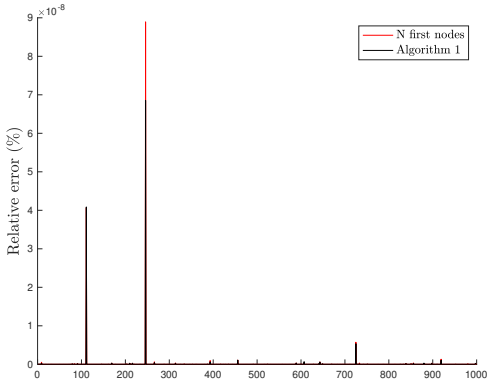


Figure 6.7: Relative error at each iteration when sampling with Algorithm 1 and when sampling the N first nodes when the system is stable in example 6.3.1

it seems reasonable to sample as many nodes at the first time step as there are stable eigenvalues and as many nodes in the last time step as there are unstable eigenvalues in order to maximize the minimum eigenvalue of $\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)}$. Again, let us emphasize that this assumption is strictly qualitative. It is based on the fact that the eigenvalues and the singular values of a matrix have "the same size" and we also consider that adding a row of a "small" matrix to an existing matrix adds a "small" singular value while adding a row of a "large" matrix adds a "large" singular value. In this case, on one hand, the stable eigenvalues will become smaller and smaller with the increasing power of $\boldsymbol{\Lambda}$ in (6.13) so we better sample at the beginning in order to maximize the minimum singular value. On the other hand, the unstable eigenvalues will become larger and larger and we better sample at the end in order to maximize the minimum singular value.

Example 6.3.2. We generate 1000 different time-varying graph signals $X \in \mathbb{C}^{N\times T}$ issued from a non-periodic dynamical system of the form $x_{t+1} = \boldsymbol{M}x_t$ such that the system is asymptotically unstable (i.e. $\rho(\boldsymbol{M}) > 1$). The matrix $\boldsymbol{M}$ is generated randomly at each iteration such that it always verifies $\rho(\boldsymbol{M}) > 1$. Again, we take $N = 10$ and $T = 20$. At each iteration, we sample 10 nodes using Algorithm 1. We run this experiment multiple times with various proportions of stable and unstable eigenvalues. Figure 6.8 shows which nodes were sampled over all the iterations. We can see that for each experiment, we have the same proportion of nodes sampled at the first 2 time instants as the proportion of stable eigenvalues and the same proportion of nodes sampled at the last 2 time instants as the proportion of unstable eigenvalues. Figure 6.9 shows the relative reconstruction error committed at each iteration when sampling N nodes with Algorithm 1 and when sampling N nodes empirically, i.e. if there are m stable eigenvalues and $N - m$ unstable eigenvalues in the system, we sample the m first nodes and the $N - m$ last nodes. We can see that the error committed when sampling nodes empirically is similar as the one committed with Algorithm 1 indicating that sampling as much nodes at the beginning as there are stable eigenvalues and as much nodes at the end as there are unstable eigenvalues gives a good approximation to the optimal sampling set.

△

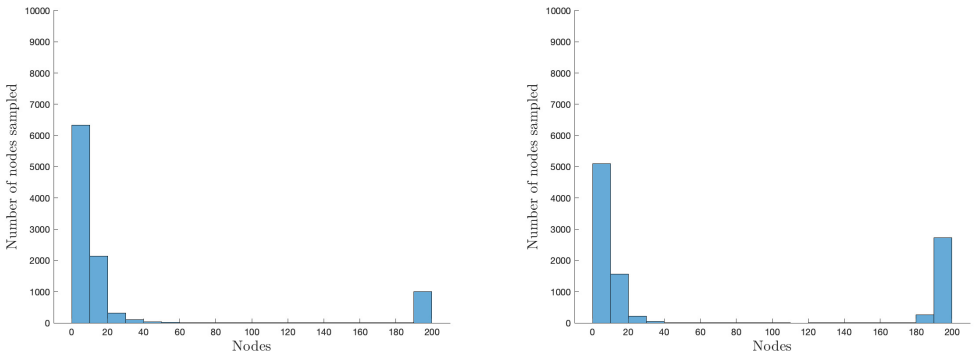
6.3.3 Summary

To summarize the empirical intuition explained above and the numerical results of examples 6.3.1 and 6.3.2, we state the following conjecture.

Conjecture 6.3.1. *Given a non periodic time-varying graph signal $X \in \mathbb{C}^{N\times T}$ whose columns satisfy $x_{t+1} = \boldsymbol{M}x_t \in \mathbb{C}^N$ with $t \in \{0, ..., T-2\}$ and x_t the t^{th} column of X, we denote by m the number of stable eigenvalues of $\boldsymbol{M}$ and by $N-m$ the number of unstable eigenvalues. The stable eigenvalues are such that $|\lambda| < 1$ or $|\lambda| = 1$ with λ being located in a one-dimensional Jordan cell. The unstable eigenvalues are such that $|\lambda| > 1$ or $|\lambda| = 1$ with λ being located in a Jordan cell of dimension larger than 1. A good approximation to the optimal sampling set returned by the optimal sampling operator*

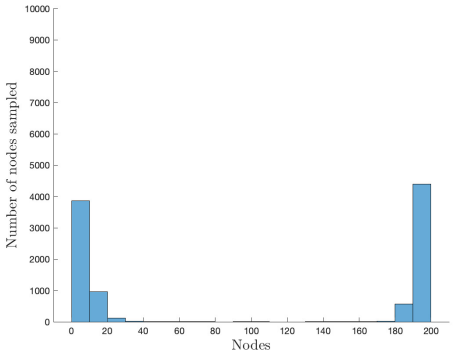
$$\boldsymbol{\Psi}^{opt} = arg\max_{\boldsymbol{\Psi}} \sigma_{min}(\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)})$$

is, when possible, to sample m nodes at $t = 0$ and $N - m$ nodes at $t = T - 1$ while still satisfying the rank theorem $rank(\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)}) = N$.

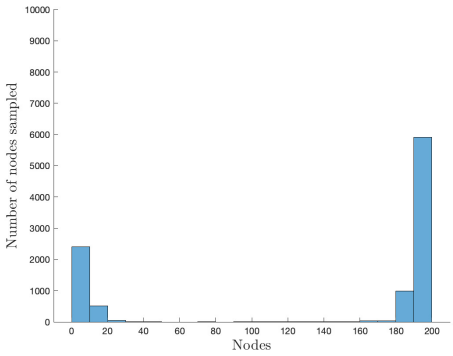


(a) 90% – 10%

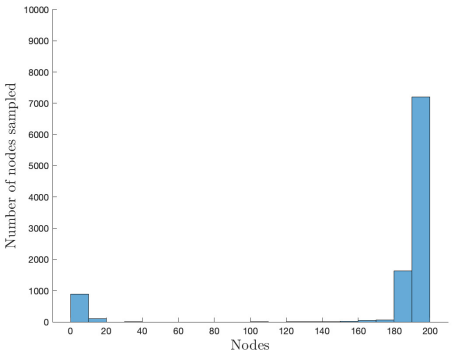
(b) 70% – 30%



(c) 50% – 50%

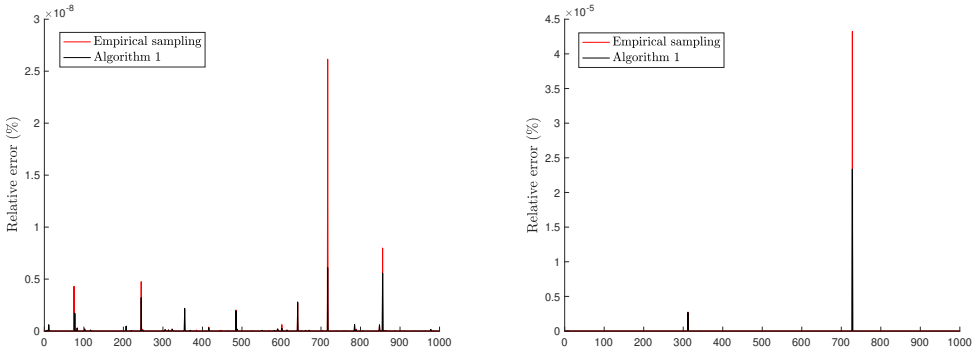


(d) 30% – 70%



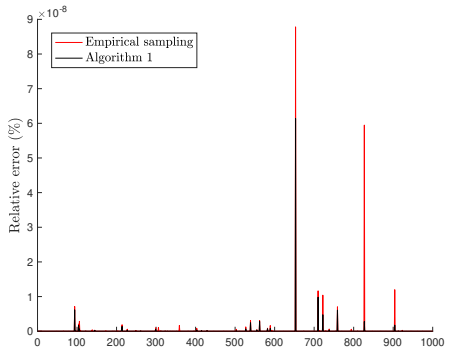
(e) 10% – 90%

Figure 6.8: Histograms of the number of nodes sampled when the system is asymptotically unstable with various proportions of stable-unstable eigenvalues in the system in example 6.3.2

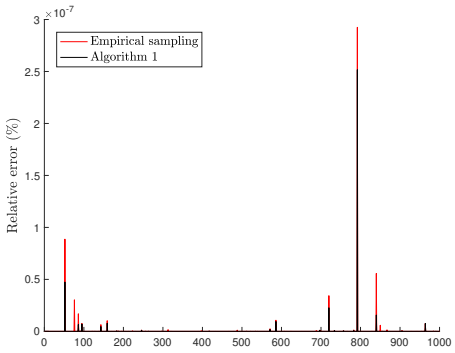


(a) 90% – 10%

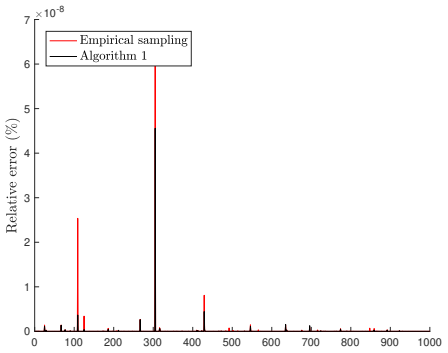
(b) 70% – 30%



(c) 50% – 50%



(d) 30% – 70%



(e) 10% – 90%

Figure 6.9: Relative error at each iteration when sampling with Algorithm 1 and when sampling empirically when the system is asymptotically unstable with various proportions of stable-unstable eigenvalues in the system in example 6.3.2

6.4 Summary

Given a non-periodic time-varying graph signal $X \in \mathbb{C}^{N \times T}$ whose columns satisfy

$$x_{t+1} = \boldsymbol{M} x_t \in \mathbb{C}^N,$$

with $t \in \{0, ..., T-2\}$, we showed in section 6.1 that X is at most N-bandlimited when using $\bar{\boldsymbol{M}}$ as graph shift.
A valid sampling operator $\boldsymbol{\Psi}$ should select nodes such that $rank(\boldsymbol{\Psi}\bar{\boldsymbol{V}}_{(N)}) = N$ with $\bar{\boldsymbol{V}}$ the eigenvector matrix of $\bar{\boldsymbol{M}}$. There are often multiple sampling sets possible that satisfy this condition. We showed in section 6.2 how some coloring rules on the dynamical graph G_D issued from $\boldsymbol{M}$ could give us some insights about the validity of a particular sampling set.
Finally, an optimal sampling operator minimizing the effect of noise should maximize the minimum singular value of $\bar{\boldsymbol{V}}_{(N)}$. We explained in section 6.3 by an argument about stability why it seems reasonable to assume that it is best to sample as many nodes at the beginning as there are stable eigenvalues of $\boldsymbol{M}$ and as many nodes at the end as there are unstable eigenvalues (while satisfying the rank theorem).

Chapter 7

Experimental results : SIR model

Now that we have developed some theory about time-varying graph signals issued from dynamical systems, we will study how this can be applied on some synthetic (but inspired from a real-life application) data. We will simulate a time-varying signal issued from the SIR compartmental model.

7.1 Experimental setup

In this section, we consider data that are created from the SIR model which is used in epidemiology to model the spread of an infectious disease [Hethcote, 2000]. In this model, we consider three compartments which are S for the number of susceptible individuals, I for the number of infectious and R for the number of recovered or deceased individuals. The three variables represent the number of people in each compartment at a particular time and vary over time. The SIR system without birth and death is expressed as follows :

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta IS}{\bar{N}}, \\ \frac{dI}{dt} &= \frac{\beta IS}{\bar{N}} - \gamma I, \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{7.1}$$

with $\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0$ and $S(t) + I(t) + R(t) = \bar{N}$. The parameter β is the rate at which an infected person infects a susceptible person while γ is the rate at which infected people recover from the disease. The linearization of (7.1) around the equilibrium point with zero infectious individual ($S^* = \bar{N}, I^* = 0, R^* = 0$) is

given by

$$\begin{aligned} \frac{dS}{dt} &= -\beta I, \\ \frac{dI}{dt} &= (\beta - \gamma) I, \\ \frac{dR}{dt} &= \gamma I, \end{aligned} \tag{7.2}$$

Looking at (7.2), we see that the dynamics of the epidemic will depend on the quantities β and γ and more precisely on their ratio $R_0 = \frac{\beta}{\gamma}$ called the basic reproduction number. If $R_0 > 1$, then $\frac{dI}{dt}(0) > 0$ and we have an epidemic, i.e. the fixed point $(S^* = \bar{N}, I^* = 0, R^* = 0)$ is unstable since a small number of initially infected people will lead to a growing number of infected people.

The spatial network that we will consider is composed of nodes representing various airports across the United States. A directed edge connects one node to another if there is a flight from this airport to the other and the weight on each edge indicates the number of passengers between two airports in the year 2010 [KONECT, 2017] [Opsahl, 2011]. To reduce the number of airports considered in the United States, only the 40 airports with the most outgoing passengers were kept. This gives us a network containing 40 nodes distributed across the United States. The network can be seen on Figure 7.1 where the edges between the airports are not shown for the sake of clarity. To avoid any confusion in what follows, we denote by N the number of nodes (the number of airports) in the network and by $\bar{N}$ the number of individuals considered in the signal issued from the SIR model.

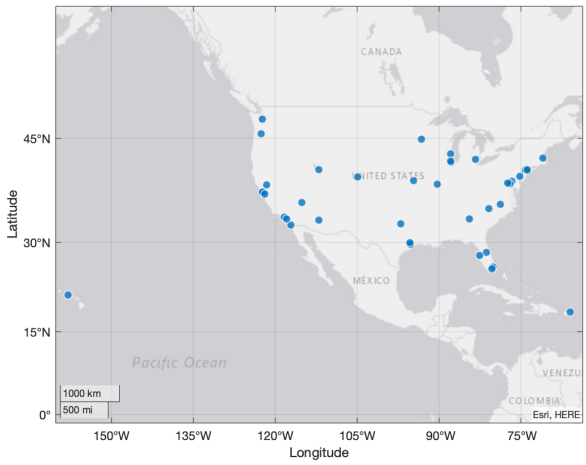


Figure 7.1: US airports network

Since we work on a network with multiple regions (one around each airport), we consider the quantities of susceptible, infectious and recovered individuals separately

on each node of the network. We will hence use a multi-region variant of the SIR model [Brugnano and Iavernaro, 2020]. Let us define

$$s = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}, \quad i = \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix}, \quad r = \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix}, \quad \bar{n} = \begin{bmatrix} \bar{n}_1 \\ \vdots \\ \bar{n}_N \end{bmatrix}$$

as the vectors of susceptible, infectious, recovered and total number of people on each node (each region around an airport). These quantities vary in time. The basic model (7.1) becomes

$$\begin{aligned} \frac{ds}{dt} &= -\boldsymbol{B}s \odot i \oslash \bar{n} + \boldsymbol{T}s, \\ \frac{di}{dt} &= \boldsymbol{B}s \odot i \oslash \bar{n} - \boldsymbol{C}i + \boldsymbol{T}i, \\ \frac{dr}{dt} &= \boldsymbol{C}i \end{aligned} \tag{7.3}$$

where $\odot$ denotes the Hadamard (i.e., component-wise) product, $\oslash$ denotes the Hadamard division and

$$\boldsymbol{B} = \begin{bmatrix} \beta_1 & & \\ & \ddots & \\ & & \beta_N \end{bmatrix} \quad \text{and} \quad \boldsymbol{C} = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_N \end{bmatrix}$$

are the matrices with the infection and removal rates. We see that these matrices allow us to define different dynamics for the epidemic following the different regions in the United States for example. The last term in the first two equations of (7.3) models the transfer of people between the different airports. Matrix $\boldsymbol{T}$ is defined as follows :

$$\begin{aligned} \boldsymbol{T}_{ij} &= \frac{\boldsymbol{A}_{ji}}{365\, p_j}, \ i \neq j \\ \boldsymbol{T}_{ii} &= \frac{-\sum_{j \neq i} \boldsymbol{A}_{ji}}{365\, p_i} \end{aligned} \tag{7.4}$$

where $\boldsymbol{T}_{ij}$ is the transfer coefficient from airport j to airport i, $\boldsymbol{A}$ is the adjacency matrix of the airport network and p_i is the population in the region of airport i. The non-negative non-diagonal coefficients $\boldsymbol{T}_{ij}$ allow to model the arrivals in a region i from a region j and the negative diagonal coefficients $\boldsymbol{T}_{ii}$ allow to model all the departures from a region i. Since $\sum_{i=1}^{N} \boldsymbol{T}_{ij} = 0$, $j = 1, ..., N$, we can then verify that the conservation property is satisfied,

$$\sum_{i=1}^{N} \left[\frac{ds_i}{dt} + \frac{di_i}{dt} + \frac{dr_i}{dt} \right] = 0$$

hence

$$\sum_{i=1}^{N} \left[s_i(t) + i_i(t) + r_i(t) \right] = \bar{N}$$

The signal we will consider on the network is the number of susceptible and infectious in each region. The notations used will be as follows : $x_t = (s(t), i(t))^T \in \mathbb{C}^{2N}$ is the signal at instant t. We can see this signal as being composed of two components on each node k: $s_k(t)$ and $i_k(t) \in \mathbb{C}$. We do not consider the quantity of recovered people in the signal to avoid increasing too much the computational load but this quantity could also be added without modifying the results obtained in what follows. Let us note that our signal is bi-variate on a network with 40 nodes thus each node consists in two nodes (one for each variable of the signal).

Since we consider a discrete time signal that has to follow the dynamic of (7.3), we must first linearize and discretize this system around an equilibrium point. The equilibrium point we will consider is the case with zero infectious individual, $(s^* = \bar{p}, i^* = 0, r^* = 0)$ with $\bar{p}$ the vector of the initial population in each region. The linearization and discretization (taking $\Delta t = 1$) is then given by

$$\begin{aligned} s(t+1) &= s(t) - \boldsymbol{B}i(t) + \boldsymbol{T}s(t), \\ i(t+1) &= i(t) + \boldsymbol{B}i(t) - \boldsymbol{C}i(t) + \boldsymbol{T}i(t), \\ r(t+1) &= r(t) + \boldsymbol{C}i(t) \end{aligned} \tag{7.5}$$

Our time varying graph signal thus follows the dynamic given by

$$\begin{aligned} x_{t+1} = \begin{bmatrix} s(t+1) \\ i(t+1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{I}_N + \boldsymbol{T} & -\boldsymbol{B} \\ \boldsymbol{0}_N & \boldsymbol{I}_N + \boldsymbol{B} + \boldsymbol{T} - \boldsymbol{C} \end{bmatrix} \begin{bmatrix} s(t) \\ i(t) \end{bmatrix} \\ &= \boldsymbol{M}x_t \end{aligned} \tag{7.6}$$

The signal can be generated with an initial condition being a small perturbation $I_p \in \mathbb{C}^N$ from the equilibrium point, i.e. $x_0 = (s(0), i(0))^T = (\bar{p} - I_p, I_p)$. In what follows, I_p was chosen as a uniformly distributed random perturbation in the interval $[0, 300]$.

For each experiment, we will consider two different matrices as graph shift :

Classical graph shift The adjacency matrix $\bar{\boldsymbol{A}}_{adj}$ of the complete network.

Dynamical graph shift The dynamical matrix $\bar{\boldsymbol{M}}$ of equation (6.2).

Remark. Since the signal is of size $2N$ (it is bi-variate) instead of N, we have to define the adjacency matrix of the network accordingly. A natural way to represent the adjacency matrix of a multivariate signal is to take the Cartesian product between the path graph and the basic adjacency matrix. Since we are dealing with a two-dimensional signal on the spatial network of the US airports with adjacency matrix $\boldsymbol{A}$, we will consider the following matrix as the classical graph shift:

$$\bar{\boldsymbol{A}}_{adj} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{I}_N \\ \boldsymbol{I}_N & \boldsymbol{A} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \oplus \boldsymbol{A}.$$

For each experiment, we then compare the results obtained with both the classical and the dynamical graph shifts. The first experiment is the 'ideal' case where the signal exactly follows the dynamic given by (7.6). In the second experiment, we add noise to the signal. In the last experiment, the signal we sample is issued from the nonlinear system (7.3). We will work as follows :

Frequency content We compare the frequency content of the signal in both graph shifts by computing the entropy of the Fourier transforms. The entropy is given by

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i \tag{7.7}$$

where n is the number of frequencies in the Fourier domain and P_i is the frequency content of the signal in the i^{th} frequency (normalized by the total frequency content of the signal in all frequencies).

Relative reconstruction error with varying number of sampled nodes We compare the relative reconstruction error in function of the number of nodes sampled via Algorithm 2. We sample from 1 to $2NT$ nodes with T the number of time steps over which the signal is generated.

Relative reconstruction error with varying parameter In the second experiment, we compare the relative reconstruction error in function of the noise added when sampling $2N = 80$ nodes. In the third experiment, we compare the relative reconstruction error in function of the number of time steps over which the signal is generated when sampling $2N = 80$ nodes.

Sampled nodes We look at which nodes are preferably sampled by Algorithm 2 when sampling $2N = 80$ nodes.

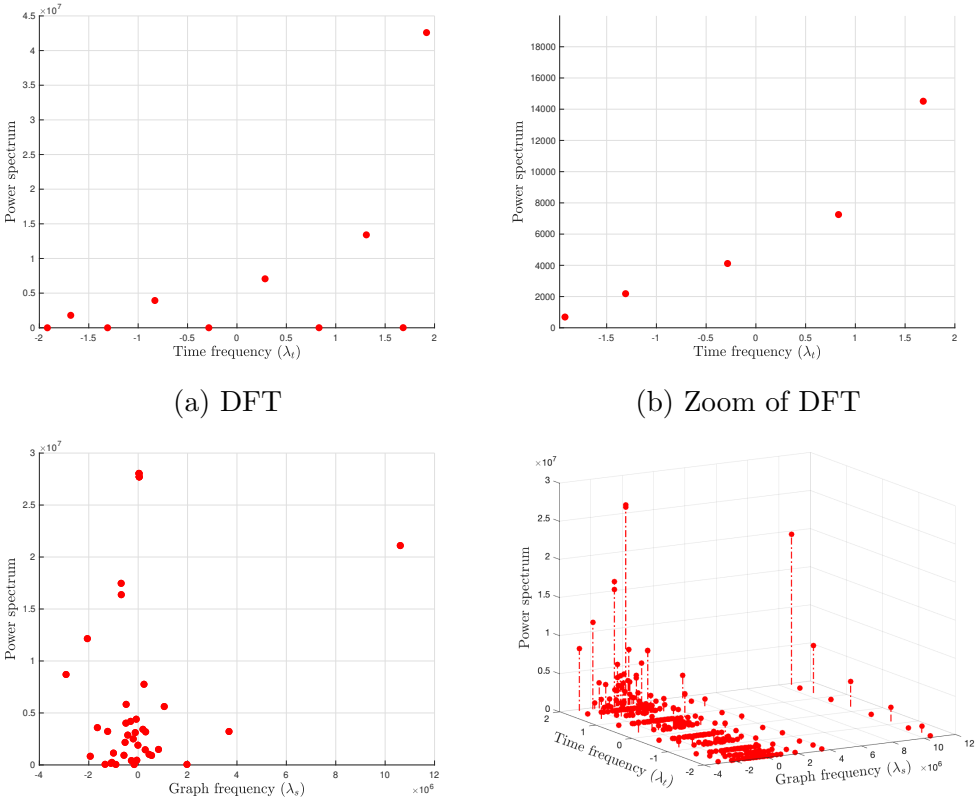
Computation time We compare the experimental computation times of Algorithm 2 for both graph shifts.

7.2 First experiment : comparison between classical and dynamical graph shift

First of all, we will compare the results obtained with the classical graph shift and the dynamical graph shift developed in section 6.1. Time will be represented with an undirected path graph. The signal spans over 10 time steps. We thus have a spatial network with $2N = 80$ nodes (40 airports and two variables on each airport) and a time network with $T = 10$ nodes.

Looking at the different Fourier transforms of the signal on Figure 7.2, we can see that it is neither bandlimited in the graph frequency domain nor in the time

frequency domain and it is also not bandlimited in the joint frequency domain. However, we see that some Fourier components carry much more information about the signal than others, together in the DFT (Discrete Fourier Transform), GFT (Graph Fourier Transform) and JFT (Joint Fourier Transform). The signal is approximately bandlimited. This can also be seen when computing the entropy (equation (7.7)) of the different transformations (Table 7.1) which is lower than the maximum possible entropy. This indicates that the signal is better explained by some of the eigenvectors of the graph shift than by others.



(a) DFT (b) Zoom of DFT

(c) GFT (d) JFT

Figure 7.2: First experiment : power spectrum of the signal obtained by the DFT, GFT and JFT. The signal is issued from (7.6) with $T = 10$ and $N = 40$.

Now from the way we built the signal, we know that it is $2N$-bandlimited when using the Fourier basis from $\bar{\boldsymbol{M}}$ in (6.2). Hence it is possible to sample $2N$ nodes from the joint network and get perfect recovery of the whole signal. We can also compute the (joint) entropy (Table 7.1) which shows that the signal is even more concentrated on some eigenvectors of the dynamical graph shift than with the classical graph shift.

We will compare both graph shifts in terms of the relative reconstruction error when we increase the number of sampled nodes. In the classical case, we can choose

	DFT	GFT	JFT
Max entropy	$\log_2(10)$ $= 3.3219$	$\log_2(80)$ $= 6.3219$	$\log_2(800)$ $= 9.6439$
Classical graph shift - Entropy	1.6042	5.2828	6.8866
Dynamical graph shift - Entropy	/	/	5.8699

Table 7.1: First experiment : entropy of the different Fourier transforms. The signal is issued from the linear system (7.6) with $T = 10$ and $N = 40$.

to sample time and space separately or jointly. In this case, since the signal is only approximately bandlimited, we choose to sample time and space jointly. This will ease the comparison with the dynamical case. Let us just note that sampling in space and time separately would allow us to reduce the computational load (when selecting the sampling set with Algorithm 1 or 2) but we would lose flexibility in the different possible sampling sets which could only lead to a possibly higher relative error when using the classical graph shift.

Figure 7.3 shows the relative error in function of the number of sampled nodes with both graph shifts. First of all, we observe that the error drops to zero for the dynamical graph shift when the number of sampled nodes is greater than $80 = 2N$. This is in accordance with the theory. The error with the classical graph shift decreases as we sample more and more nodes but we have to sample more than 250 nodes (about one third of the total number of nodes) in order to get a relative error of less than 10% which shows that this graph shift is inappropriate for such signals compared to the dynamical graph shift.

We can also look at which nodes are preferably sampled by the sampling algorithm (Algorithm 2) used (when sampling 80 nodes). Figure 7.4 shows a histogram of the number of nodes sampled from both quantity s (susceptible people) and i (infectious people) and another histogram of the number of nodes sampled at each time instant. In this case, we see that the majority of the 80 nodes are sampled from the quantity s and we can observe an almost even division between time 0 and time 9. Looking at the eigenvalues of the dynamic matrix $\boldsymbol{M}$, we observe 23 unstable eigenvalues ($|\lambda| > 1$) which partly explains the number of nodes sampled at the last time step as discussed in section 6.3.

Finally, let us note that although the sampling algorithm used with both graph shifts is the same (Algorithm 2) with inputs and outputs of the same size, the experimental computation times observed vary a lot between the two methods. Indeed, as can be seen on Figure 7.5, the computation time to select the sampling set when using the dynamical graph shift is systematically smaller than with the classical graph shift. This observation also holds for the second and third experiment.

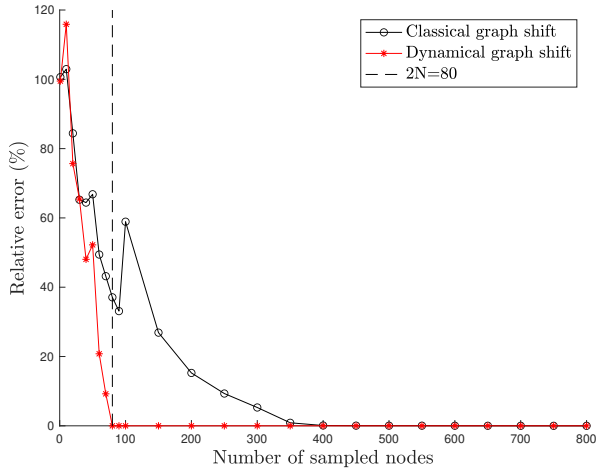
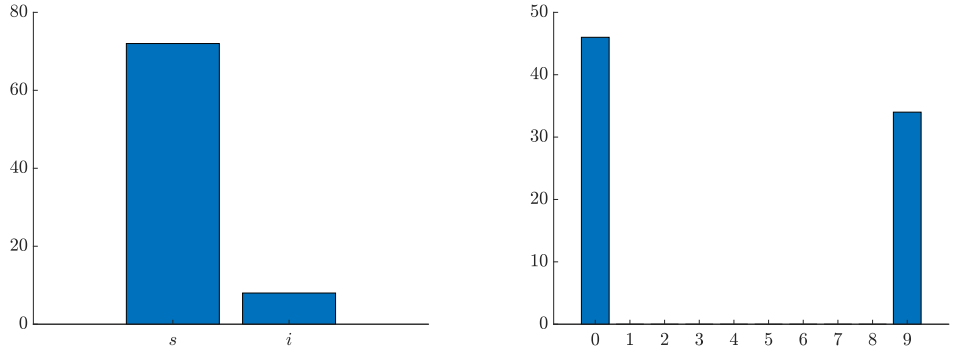


Figure 7.3: First experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts. The signal is issued from (7.6) with $T = 10$ and $N = 40$.



(a) Quantity of s nodes versus quantity of i nodes

(b) Quantity of nodes at the different time steps

Figure 7.4: First experiment : histograms of the number of nodes sampled. Left : comparison between the quantity of nodes sampled with value of s and with value of i. Right : comparison between the different time steps.

We also ran this first experiment on the US airport network with $N = 160$ airports. The results are similar to the ones obtained in this section and are presented in the appendix A.1.

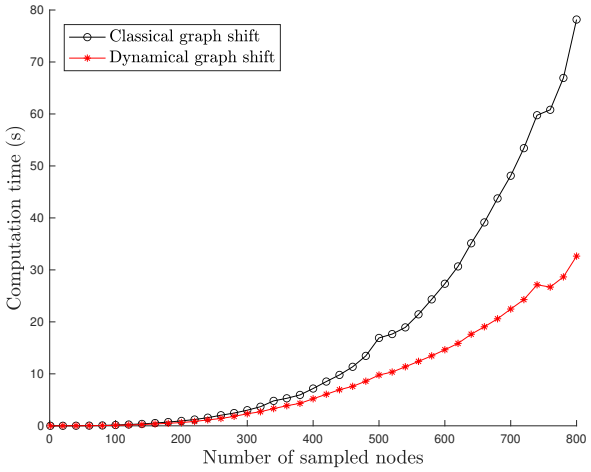


Figure 7.5: First experiment : evolution of the computation time of Algorithm 2 with the number of sampled nodes with the classical and the dynamical graph shifts.

7.3 Second experiment : additive noise in the signal

In the second experiment, we add some noise to the signal. We will conduct two different experiments here. In the first one, we add a constant noise and we sample more and more nodes in the network. In the second one, we add more and more noise while still sampling $2N = 80$ nodes.

In general, we can consider that the signal present in the real world is

$$X_{real} = X_{SIR} + N_{model}$$

where X_{SIR} is the signal issued from the SIR dynamical system (an approximation of the reality) and N_{model} is noise issued from the model (the model is an approximation of what happens in real life where the signal is usually not deterministic). X_{SIR} represents the deterministic part of the signal while N_{model} is the non-deterministic part. Then, the signal we measure on the nodes of the network is given by

$$X_{measured} = X_{real} + N_{measurements} = X_{SIR} + N_{model} + N_{measurements}$$

where $N_{measurements}$ is noise issued from the measurements (error in the measurements, etc). In this section, the noise we add is considered as being noise issued from the model, i.e., noise representing the non-deterministic part of the real signal. We ignore the noise issued from the measurements as the impact of this noise is already minimized by the sampling algorithm used (section 2.1). The signal we want to recover is the true signal X_{real} while the dynamical graph shift was designed to

perfectly recover the deterministic part of this signal X_{SIR}. In this section, we will thus study the impact of the non-deterministic part of the signal on the reconstruction performance when using the dynamical graph shift (and we still compare these performances to the ones obtained with the classical graph shift).

First, we simulate the signal over 10 time steps. The signal X_{SIR} taken is the same as the one used in the first experiment. We then add white Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ such that the signal-to-noise (SNR) ratio given by

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{X_{SIR}}}{P_{noise}} \right) \tag{7.8}$$

is constant, with $P_{X_{SIR}}$ and P_{noise} the power of the signal and of the noise respectively. Let us note here that the conservation property

$$\sum_{i=1}^{N} [s_i(t) + i_i(t) + r_i(t)] = \bar{N}$$

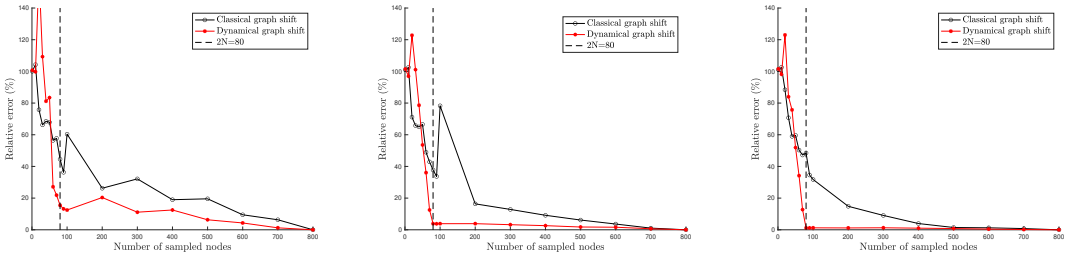
is not verified anymore in this case. We then look at how the relative error evolves when sampling more and more nodes with a constant added noise.

The entropy of the joint Fourier transform of the noisy signal (with SNR of 20, 30 and 40 dB) and of its noise-free version are given in Table 7.2. We can see that the entropy is higher for the noisy signal than without noise for both graph shifts. The entropy is again smaller with the dynamical graph shift and decreases faster when we decrease the noise added. We should again get better results than with the classical graph shift.
The results are shown on Figure 7.6 for various SNR (20, 30 and 40 dB). Concerning the dynamical graph shift, we observe a sudden drop in the relative reconstruction error when sampling $2N = 80$ nodes. We do not reach a zero error as in the ideal case of the first experiment in section 7.2 but we still observe the effect of this theoretical bound. Then, we can see that increasing the number of nodes above this value does not have a great impact on the error. Indeed, the relative reconstruction error (for 20 dB) does not really decrease before sampling 500 nodes (about two third of the total number of nodes). This bound of $2N$ nodes thus seems to remain the optimal choice concerning the number of nodes to sample with the dynamical graph shift as far as the trade off between the number of nodes to sample and the magnitude of the relative error is concerned. Concerning the error with the classical graph shift however, we can observe a real decreasing trend but just as in section 7.2, we have to sample more nodes than with the dynamical graph shift (almost every node) in order to reach an acceptable threshold for the error. In fact, we observe the same trend in these figures than in Figure 7.2.

We conclude this experiment by observing that the noise only has a perceptible

		Max entropy	JFT entropy	
			Noise-free signal	Noisy signal
Classical graph shift	20 dB	9.6439	6.8866	7.4803
	30 dB			7.1346
	40 dB			6.9861
Dynamical graph shift	20 dB		5.8699	7.3797
	30 dB			6.6272
	40 dB			6.2242

Table 7.2: Second experiment : entropy of the joint Fourier transform when noise is added in the signal issued from the linear system (7.6) with $N = 40$ and $T = 10$.



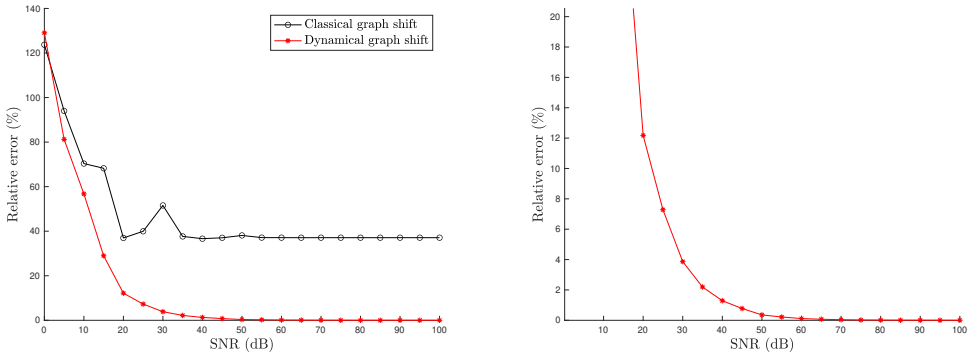
(a) 20 dB (b) 30 dB (c) 40 dB

Figure 7.6: Second experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts. The signal is issued from (7.6) with $T = 10$ and $N = 40$ with added noise at constant signal to noise ratio (20, 30 and 40 dB).

impact on the dynamical graph shift which still performs well when the SNR is not too low.

Next, we add white Gaussian noise such that the SNR is increasing and goes from 0 to 100 dB. With both graph shifts, the number of sampled nodes is $2N = 80$. Figure 7.7 shows the relative error in function of the SNR. We can see that our dynamical graph shift does not perform well for a small signal to noise ratio but with a SNR of above 30 dB, we get a reconstruction error of less than 5% and this reconstruction error continues to decrease as the SNR increases. Our dynamical graph shift still performs a lot better than the classical graph shift which has a relative error of about 40% whatever the signal to noise ratio considered. In fact, the noise does not seem to have any impact there since it already had a relative error of about 40% when sampling 80 nodes on the noise-free signal (see Figure 7.2).

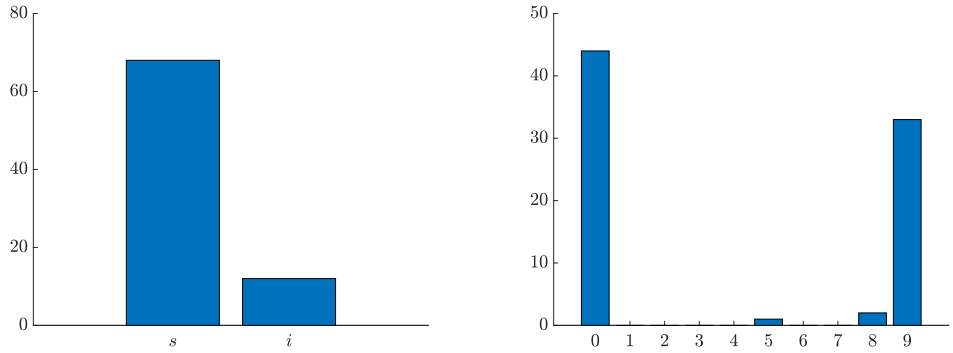
Finally, we can compare the sampling set when sampling 80 nodes in this experiment to the sampling set of the experiment in section 7.2 to see if the noise has an impact on the optimal sampling set. The sampling set is almost exactly the same



(a) General view (b) Zoom

Figure 7.7: Second experiment : evolution of the relative error with the SNR with the classical and the dynamical graph shifts when the number of sampled nodes is $2N = 80$. The signal is issued from (7.6) with $T = 10$ and $N = 40$ with added noise.

when adding noise has can be seen on Figure 7.8. Again, the majority of the 80 nodes are sampled from the quantity s and we can observe an almost even division between time 0 and time 9. We can observe that some nodes are sampled at time 5.



(a) Quantity of s nodes versus quantity of i nodes (b) Quantity of nodes at the different time steps

Figure 7.8: Second experiment : histograms of the number of nodes sampled. Left : comparison between the quantity of nodes sampled with value of s and with value of i. Right : comparison between the different time steps.

We also ran this second experiment on the US airport network with $N = 160$ airports. The results are similar to the ones obtained in this section and are presented in the appendix A.2.

7.4 Third experiment : introduction of non-linearities

In the third experiment, we simulate the signal with the nonlinear dynamical system (7.3) but we still build $\bar{\boldsymbol{M}}$ from the linearization (7.5) of this system. We will again compare the results obtained with the classical graph shift and the dynamical graph shift. In this experiment, we will increase the number of time steps where we observe the signal. Indeed, in this case, we expect the linearization (7.5) of the model (7.3) to be a good approximation when T is small but to become inaccurate when T increases. We will thus compare the performances of both graph shifts when T is small and when T is large.

First of all, we again simulate the signal over 10 time steps. In this case, the linear approximation stays accurate, we are still in the linear regime and the results obtained are comparable to the ones obtained in the first experiment in section 7.2. These are presented in the appendix A.3.

Then, we simulate the signal over 40 time steps (the total number of nodes is now 3200). In this case, the linear approximation is not accurate anymore, we have entered the nonlinear regime. The entropy of the joint Fourier transform of this signal and of its linear approximation are given in Table 7.3. We can see that the difference of entropy between the real signal and its linear approximation is not negligible with the dynamical graph shift. The real signal is less sparse in this representation than its linear approximation. This was expected since we know that the linear approximation is only valid for a small time lapse. We thus expect to get worse results concerning the reconstruction error in this case. Nevertheless, we can still observe that the entropy of the signal with the dynamical graph shift is smaller than when using the classical graph shift so we should still get better results.

Looking at the relative reconstruction error, we directly see on Figure 7.9 that it is higher than in the 'ideal' case of the first experiment with both graph shifts. We can still observe a large drop between 1 and 80 sampled nodes with the dynamical graph shift but the relative error is not negligible anymore. We now have to sample more than 150 nodes to get an error of less than 10%. This confirms what we expected, namely that the linearization of the model (7.3) is a good approximation when T is small (for example $T = 10$) but it becomes less accurate when T increases. However, even with a relatively large T, we are still able to get an acceptable reconstruction error (for example about 5%) with a relatively small amount of sampled nodes (here about 300 nodes which represents one tenth of the total number of nodes). And still, these results are better than the ones obtained with the classical graph shift where we have to sample more than 2000 nodes (two third of the total number of nodes) to get an error of less than 10%.

	Max entropy	JFT entropy	
		Nonlinear signal	Linear signal
Classical graph shift	11.6439	8.9582	9.7393
Dynamical graph shift		7.2128	5.9412

Table 7.3: Third experiment : entropy of the joint Fourier transform. The signal is issued from the nonlinear dynamical system (7.3) with $N = 40$ and $T = 40$.

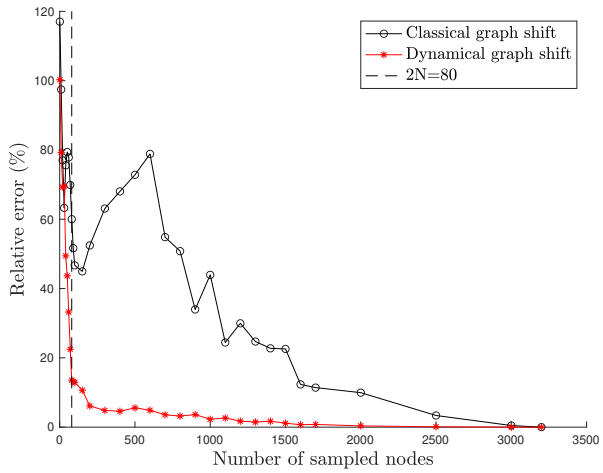
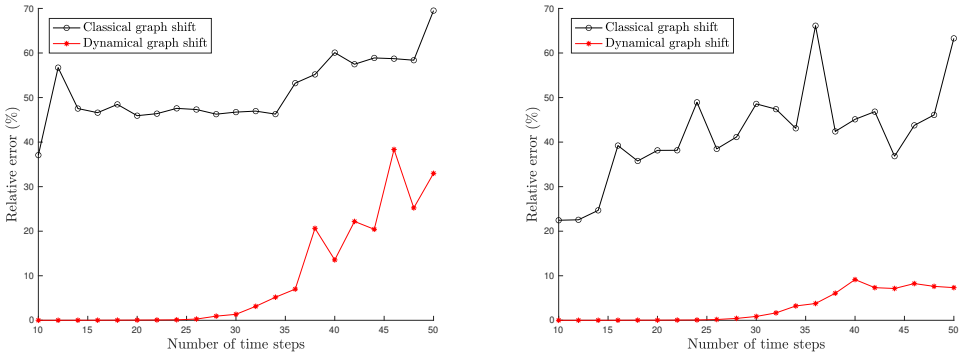


Figure 7.9: Third experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts when $T = 40$. The signal is issued from the nonlinear dynamical system (7.3) with $N = 40$.

Finally, let us look at the evolution of the error when we always sample $2N = 80$ nodes but the signal is simulated over more and more time steps. This will allow us to assess to which extent we can consider that our linearized model is a good approximation to the original model. On Figure 7.10a, we can see that when sampling 80 nodes and depending on the acceptable threshold for the application, using the dynamical graph shift gives a reconstruction error of less than 10% up until about 35 time steps while the classical graph shift does not give any acceptable results when sampling as few nodes as we did in this experiment.

It is worth noting that by doubling the number of sampled nodes to 160, we can get a reconstruction error of less than 10% even when $T = 50$ as can be seen on Figure 7.10b while the classical graph shift still gives a relative error of more than 20% for small time lapses and more than 40% for large time lapses.



(a) When sampling $2N = 80$ nodes (b) When sampling 160 nodes

Figure 7.10: Third experiment : evolution of the relative error with the number of simulated time steps with the classical and the dynamical graph shifts when the number of sampled nodes is $2N = 80$. The signal is issued from the nonlinear dynamical system (7.3) with $N = 40$.

Finally, we can compare the sampling set when sampling 80 nodes in this experiment to the sampling set of the first experiment to see if the non-linearities have an impact on the optimal sampling set. In this case, we only consider the experiment over 10 time steps because the total number of nodes is then then same. In this case, the sampling set is exactly the same as in the first experiment.

7.5 Conclusion

Through the three experiments conducted in this chapter, we have seen that using the dynamical graph shift developed in section 6.1 is more efficient than using the classical graph shift such as the adjacency matrix.
In each of the experiments, the number of nodes to sample allowing for a small reconstruction error is much smaller with the dynamical graph shift than with the classical graph shift and we have seen that the theoretical bound of N nodes (N being the dimension of the signal) sufficient to get perfect reconstruction in the 'ideal' linear case without noise is also a good choice concerning the trade-off between the number of nodes to sample and the reconstruction error when noise is added to the signal. Concerning the case where the signal is issued from a nonlinear dynamical system and where the dynamical graph shift is build from the linearization of this system, our graph shift also stays competitive when we stay in the linear regime. The performances with the dynamical graph shift get worse when entering the nonlinear regime but can be improved by sampling more nodes and are still better than with the classical graph shift. Finally, the algorithm used for the sampling set selection executes faster when using the dynamical graph shift.
Let us note that we conducted the experiments in this chapter on synthetic data.

It could be interesting to assess the performances of the dynamical graph shift on some real-world data instead.

Conclusion

The sampling and reconstruction problem has already been extensively studied in the last ten years for static graph signals and more recently for time-varying graph signals. The classical GSP sampling theory uses prior knowledge about the underlying graph structure to efficiently sample these signals through the graph shift (adjacency matrix or graph Laplacian) but it does not take advantage of any prior knowledge we could have about the underlying dynamic of these signals. In this thesis, we have studied how we could use a new graph shift to exploit the knowledge we have when signals are issued from linear dynamical systems and we have showed that it improves the performances obtained in sampling and in reconstruction.

In the first part of this thesis, we reviewed the notions of graph signal processing and of sampling for arbitrary static and time-varying graph signals.

In the second part, we started from the classical theory and adapted it to the case where the time-varying signals are issued from linear dynamical systems. Our main contribution was to provide a new graph shift (both for the periodic and the non-periodic case) for the underlying network which exploits the underlying dynamic of the signal. This allowed to systematically have a sparse representation of the signal in the frequency domain and put a bound on the sufficient number of nodes to sample for perfect recovery of the whole signal. Building a dynamical graph based on the dynamic of the signal, we also showed how we could identify a valid sampling set by applying some coloring rules and explained how finding a valid sampling set was similar to the problem of finding a zero-forcing set on this graph. Finally, we showed by numerical results how the stability of the linear system had an impact on the optimal sampling strategy to apply and we explained this via some heuristic arguments.

We have seen the importance of considering such signals issued from dynamical systems. This type of signal commonly arises in real-world applications as many real-world signals are usually modeled by dynamical systems (the spread of a disease, heat diffusion processes, consensus algorithms, etc). Taking into account the prior knowledge we have on these signals through a new graph shift, we showed that we could improve the performances of sampling and of reconstruction in terms of the

number of nodes to sample and of the reconstruction error compared to the classical graph shift, making the process of sampling and reconstruction much more efficient. We think that there is still much to learn in the field of GSP for time-varying graph signals issued from dynamical systems and this could make the solutions to the various problems faced with these signals much more efficient.

Some of the issues we have gone through have only been covered a little, and there are still some open questions remaining. Indeed, in sections 5.2 and 6.2, we have talked about the coloring rule on the dynamical graph corresponding to the dynamic of the signal which allows to identify a valid sampling set for the signal. The conjecture we stated about this remains to be proved. It would also be interesting to find if an efficient algorithm exists to solve this problem and this could be found in the field of the zero-forcing set problem. Another problem that was only slightly covered is the impact of the stability of the system on the optimal sampling strategy to apply.
Let us also note that in this thesis, we only considered static networks but it often happens that the networks change with time. The underlying dynamic of the time-varying graph signal could also evolve with time and as was mentioned in chapter 7, the signal could be issued from a non-linear dynamical system. All these issues could be explored in more details and would allow to improve the performances of the sampling algorithms on real-world data.
Finally, we only looked at the sampling and reconstruction problem for those signals issued from dynamical systems but the field of graph signal processing is much larger than that. Consideration could also be given to whether other common problems in GSP such as filtering, compressed sensing or graph learning could be adapted such that the performance of the solution to these problems would be improved in the case of signals issued from dynamical systems.

Bibliography

[AIM Minimum Rank – Special Graphs Work Group, 2008] AIM Minimum Rank – Special Graphs Work Group (2008). Zero forcing sets and the minimum rank of graphs. *Linear Algebra and its Applications*, 428(7):1628–1648.

[Avron and Boutsidis, 2013] Avron, H. and Boutsidis, C. (2013). Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499.

[Berliner et al., 2017] Berliner, A., Bozeman, C., Butler, S., Catral, M., Hogben, L., Kroschel, B., Lin, J. C. H., Warnberg, N., and Young, M. (2017). Zero forcing propagation time on oriented graphs. *Discrete Applied Mathematics*, 224:45–59.

[Brugnano and Iavernaro, 2020] Brugnano, L. and Iavernaro, F. (2020). A multi-region variant of the sir model. *arXiv preprint arXiv:2003.09875*.

[Chen et al., 2015] Chen, S., Varma, R., Sandryhaila, A., and Kovačević, J. (2015). Discrete signal processing on graphs: Sampling theory. *IEEE Transactions on Signal Processing*, 63(24):6510–6523.

[Chen et al., 2016] Chen, S., Varma, R., Singh, A., and Kovačević, J. (2016). Signal recovery on graphs: Fundamental limits of sampling strategies. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):539–554.

[Halanay and Rasvan, 2000] Halanay, A. and Rasvan, V. (2000). *Stability and stable oscillations in discrete time systems*.

[Hethcote, 2000] Hethcote, W. H. (2000). The mathematics of infectious diseases. *SIAM review*, 42(4):599–653.

[KONECT, 2017] KONECT (2017). Us airports network dataset – KONECT. `http://konect.uni-koblenz.de/networks/opsahl-usairport`. Visited on 29 April 2020.

[Kurokawa et al., 2017] Kurokawa, T., Oki, T., and Nagao, H. (2017). Multi-dimensional graph fourier transform. *arXiv preprint arXiv:1712.07811*.

[Loukas and Foucard, 2016] Loukas, A. and Foucard, D. (2016). Frequency analysis of time-varying graph signals. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 346–350.

[Opsahl, 2011] Opsahl, T. (2011). Why anchorage is not (that) important: Binary ties and sample selection. `http://wp.me/poFcY-Vw`. Visited on 29 April 2020.

[Ortega et al., 2018] Ortega, A., Frossard, P., Kovačević, J., Moura, J. M. F., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

[Puy et al., 2018] Puy, G., Tremblay, N., Gribonval, R., and Vandergheynst, P. (2018). Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44(2):446 – 475.

[Sandryhaila and Moura, 2013] Sandryhaila, A. and Moura, J. M. F. (2013). Eigendecomposition of block tridiagonal matrices. *arXiv preprint arXiv:1306.0217*.

[Sandryhaila and Moura, 2014a] Sandryhaila, A. and Moura, J. M. F. (2014a). Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90.

[Sandryhaila and Moura, 2014b] Sandryhaila, A. and Moura, J. M. F. (2014b). Discrete signal processing on graphs: Frequency analysis. *IEEE Transactions on Signal Processing*, 62(12):3042–3054.

[Tzamarias et al., 2018] Tzamarias, D. E. O., Akyazi, P., and Frossard, P. (2018). A novel method for sampling bandlimited graph signals. *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 126–130.

[Varma and Kovačević, 2019] Varma, R. and Kovačević, J. (2019). Random sampling for bandlimited signals on product graphs. In *2019 13th International conference on Sampling Theory and Applications (SampTA)*, pages 1–5.

Appendix A

Additional experimental results

A.1 First experiment : comparison between classical and dynamical graph shift on a 160 nodes network

The results presented in this section were obtained with the same experimental setup as in chapter 7 on a network containing 160 nodes.

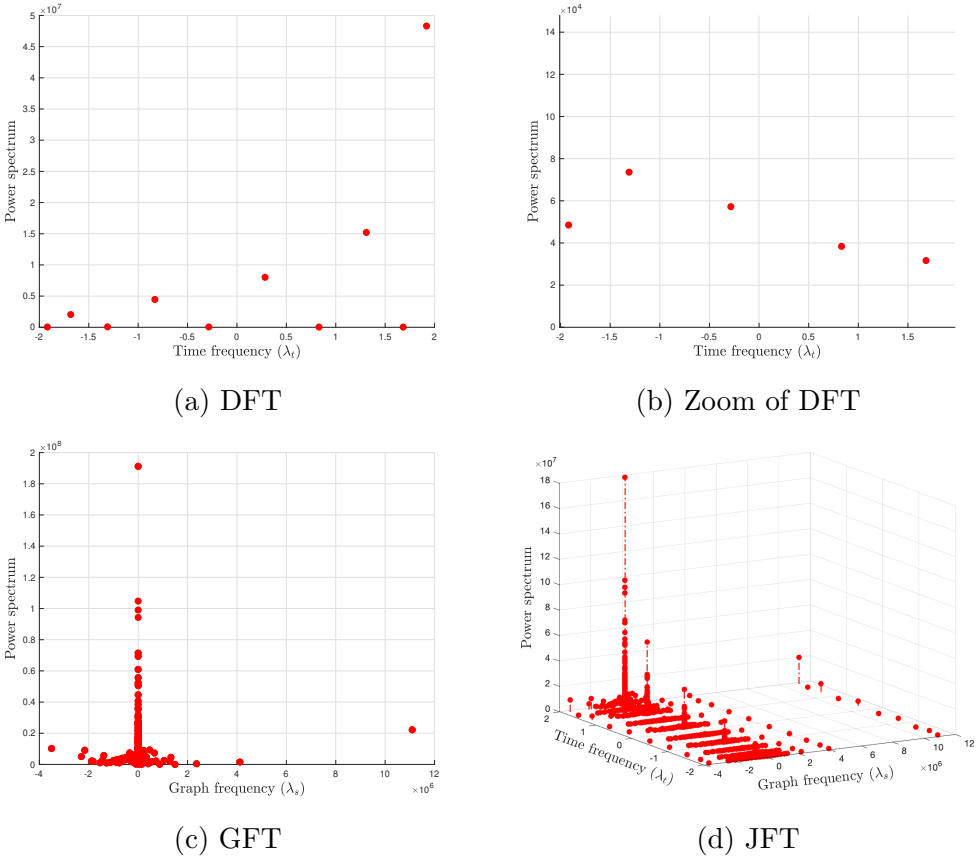
The different Fourier transforms of the signal are presented on Figure A.1 and Table A.1 presents the entropy of these different transformations. The same conclusions as in the 40 nodes case can be drawn.

	DFT	GFT	JFT
Max entropy	$\log_2(10)$ $= 3.3219$	$\log_2(80)$ $= 8.3219$	$\log_2(800)$ $= 11.6439$
Classical graph shift - Entropy	1.6322	7.2400	8.9028
Dynamical graph shift - Entropy	/	/	7.9090

Table A.1: First experiment : entropy of the different Fourier transforms. The signal is issued from the linear system (7.6) with $T = 10$ and $N = 160$.

Figure A.2 shows the relative error in function of the number of sampled nodes for both methods.

Figure A.3 shows a histogram of the number of nodes sampled from both quantity s (susceptible people) and i (infectious people) and another histogram of the number of nodes sampled at each time instant.



(a) DFT

(b) Zoom of DFT

(c) GFT

(d) JFT

Figure A.1: First experiment : power spectrum of the signal obtained by the DFT, GFT and JFT. The signal is issued from (7.6) with $T = 10$ and $N = 160$.

A.2 Second experiment : additive noise in the signal on a 160 nodes network

The results presented in this section were obtained with the same experimental setup as in chapter 7 on a network containing 160 nodes.

The entropy of the joint Fourier transform of the noisy signal (with SNR of 20, 30 and 40 dB) and of its noise-free version are given in Table A.2 and the evolution of the relative error when sampling more and more nodes at constant added noise is shown on Figure A.4 for various SNR (20, 30 and 40 dB).

Figure A.5 shows the relative error in function of the SNR.

Figure A.6 shows a histogram of the number of nodes sampled from both quantity s and i and another histogram of the number of nodes sampled at each time instant.

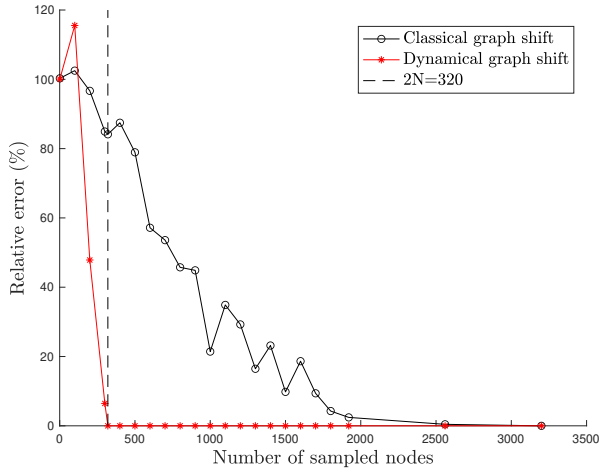
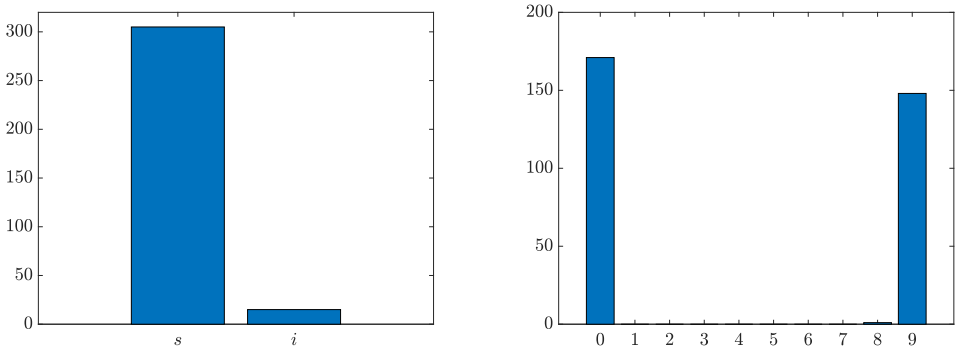


Figure A.2: First experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts. The signal is issued from (7.6) with $T = 10$ and $N = 160$.



(a) Quantity of s nodes versus quantity of i nodes

(b) Quantity of nodes at the different time steps

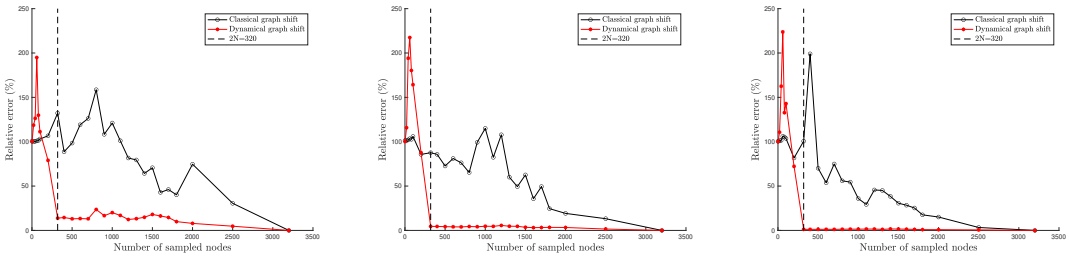
Figure A.3: First experiment : histograms of the number of nodes sampled. Left : comparison between the quantity of nodes sampled with value of s and with value of i. Right : comparison between the different time steps.

A.3 Third experiment : introduction of non-linearities with $T = 10$

We simulate the signal with the nonlinear dynamical system (7.3) but we build $\bar{\boldsymbol{M}}$ from the linearization (7.5) of this system. The signal is simulated over 10 time steps.

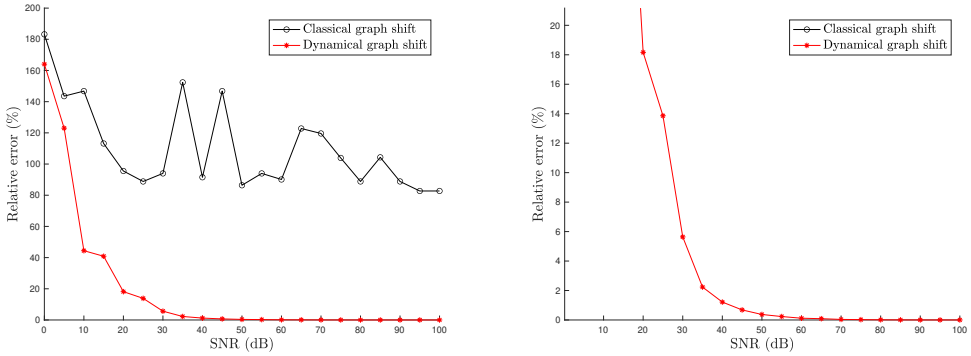
		Max entropy	JFT entropy	
			Noise-free signal	Noisy signal
Classical graph shift	20 dB	11.6439	8.9028	10.0639
	30 dB			9.3899
	40 dB			9.0800
Dynamical graph shift	20 dB		7.9090	9.3650
	30 dB			8.5822
	40 dB			8.1796

Table A.2: Second experiment : entropy of the joint Fourier transform when noise is added in the signal issued from the linear system (7.6) with $N = 160$ and $T = 10$.



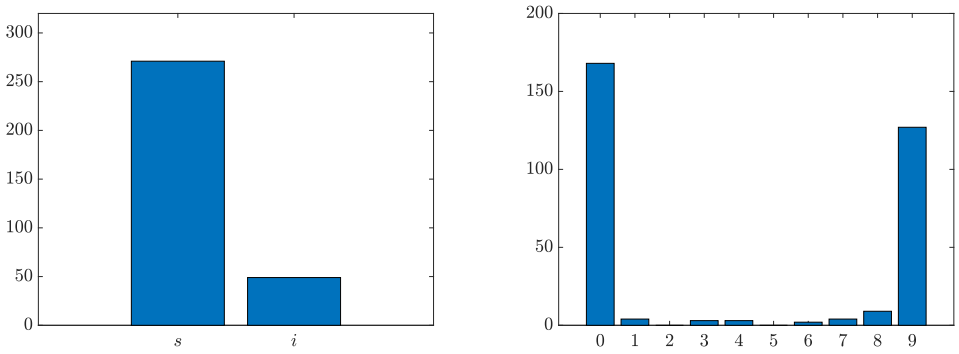
(a) 20 dB (b) 30 dB (c) 40 dB

Figure A.4: Second experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts. The signal is issued from (7.6) with $T = 10$ and $N = 160$ with added noise at constant signal to noise ratio (20, 30 and 40 dB).



(a) General view (b) Zoom

Figure A.5: Second experiment : evolution of the relative error with the SNR with the classical and the dynamical graph shifts when the number of sampled nodes is $2N = 320$. The signal is issued from (7.6) with $T = 10$ and $N = 160$ with added noise.



(a) Quantity of s nodes versus quantity of i nodes

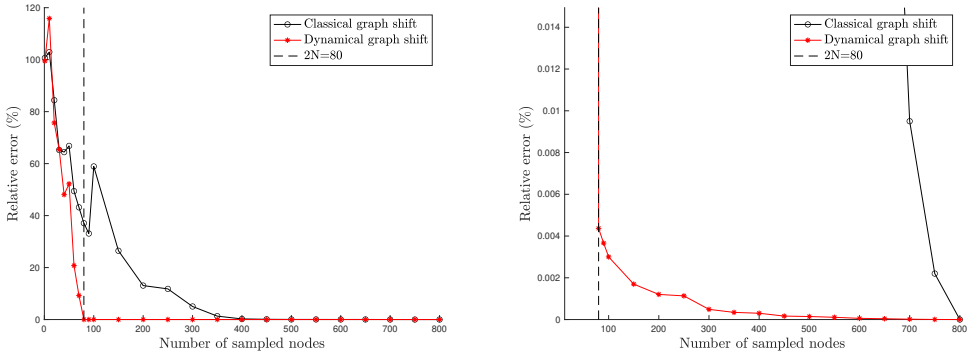
(b) Quantity of nodes at the different time steps

Figure A.6: Second experiment : histograms of the number of nodes sampled. Left : comparison between the quantity of nodes sampled with value of s and with value of i. Right : comparison between the different time steps.

The entropy of the joint Fourier transform of this signal and of its linear approximation are given in Table A.3. We can see that the difference of entropy is very small between the real signal and its linear approximation for both graph shifts so we expect the result of the reconstruction error to be similar to what we obtained in section 7.2. Also, the entropy is again smaller with the dynamical graph shift so we should again get better results than with the classical graph shift.
The results of the reconstruction error can be seen on Figures A.7a and A.7b. We can see that the dynamical graph shift is more efficient than the classical one in terms of the number of nodes we have to sample in order to reach an acceptable threshold with the relative error. We obtain a relative error of less than 10% when sampling 70 nodes and drop to an error of almost zero when sampling $2N = 80$ nodes with the dynamical graph shift while we have to sample more than 250 nodes (more than one fourth of the total number of nodes) with the classical graph shift to get an error of less than 10%.
Now, compared to the first experiment, we can see that with the dynamical graph shift the results are almost as good in this case where the signal is issued from a nonlinear dynamical system than in the case where the signal is issued from a linear system. Indeed, the relative error here becomes negligible when sampling $2N$ nodes which is the threshold to get a zero error in the linear case.

	Max entropy	JFT entropy	
		Nonlinear signal	Linear signal
Classical graph shift	9.6439	6.8878	6.8866
Dynamical graph shift		5.8708	5.8699

Table A.3: Third experiment : entropy of the joint Fourier transform. The signal is issued from the nonlinear dynamical system (7.3) with $N = 40$ and $T = 10$.



(a) General view (b) Zoom

Figure A.7: Third experiment : evolution of the relative error with the number of sampled nodes with the classical and the dynamical graph shifts when $T = 10$. The signal is issued from the nonlinear dynamical system (7.3) with $N = 40$.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**