Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

DEPARTMENT OF INFORMATION ENGINEERING

M.SC. IN CONTROL SYSTEMS ENGINEERING

Controllability of Networks: Influence of Structure and Memory

Advisor: Prof. Francesco Ticozzi

Candidate: Marco Peruzzo

Coadvisor: Prof. Giacomo Baggio

ACADEMIC YEAR 2022-2023

Graduation Date 05/09/2021

Abstract

In this work we borrow some ideas from the the theory of lifted Markov chains, which can accelerate convergence of random walks algorithms thanks to the introduced memory effects, and apply them to the control of networks of dynamical systems arranged on a line and on a grid. We lift the dynamics by enlarging each node state and discuss how to compare the effect of a control input on the lifted network and on the original one. We compute some metrics for energy-related controllability, showing that the lifted network has better controllability properties than the non-lifted one. This proves an advantage induced by the extra internal dynamics that allows for memory effects. The potential of lifts is then explored via numerical simulations for some paradigmatic examples.

Sommario

In questo lavoro prendiamo spunto dalla teoria delle Lifted Markov Chains, le quali possono accelerare la convergenza di algoritmi basati su random walks grazie agli effetti di memoria introdotti, e applichiamo alcune tecniche utilizzate in quel contesto al controllo di reti di sistemi dinamici disposti su una linea e su una griglia. In particolare, eseguiamo il lift della dinamica aumentando le variabili di stato associate a ciascun nodo e discutiamo come sia possibile confrontare l'effetto di un input di controllo sulla rete espansa e su quella originale. Calcoliamo alcune metriche per l'energia di controllo, e dimostriamo che la rete lifted ha migliori proprietà di controllabilità rispetto alla rete originale. Questo dimostra un vantaggio indotto dalla dinamica interna aggiuntiva che consente effetti di memoria. Il potenziale dell'operazione di lifting viene poi esplorato attraverso simulazioni numeriche per alcuni esempi paradigmatici.

Nomenclature

$\ker(A)$ kernel of matrix A

$\lambda_{\min}(A)$ minimum eigenvalue of a symmetric matrix A

$\|v\|_2$ ℓ_2 norm of vector v

$\mathbb{1}_n$ Column vector of dimension n whose entries are all equal to 1

$\mathbb{E}_p[x]$ Expected value under the probability distribution p of the random variable x

$\rho(A)$ Spectral radius of matrix A

$\sigma(A)$ Spectrum of matrix A

$\mathbf{1}_{[c,d]}(x)$ Indicator function, which takes value 1 if x belongs to the interval $[c, d]$, otherwise takes value 0

$\mathrm{Im}(A)$ image of matrix A

$\mathrm{trace}(A)$ trace of matrix A

$|A|$ matrix whose entry $|A|_{ij}$ is the absolute values of the entry A_{ij} of matrix A

$A^\dagger$ Conjugate transpose of matrix A

A^T Transpose of matrix A

$e_{k,n}$ k-th canonical vector of dimension n

Contents

Introduction

The importance of network theory as a tool to describe the relations between different systems has been widely highlighted in the last years. As the world is becoming every day more connected, understanding how large-scale systems can be described and exploited to reach a target result is of paramount importance from a control-theoretic perspective.
The development of network science proceeds in parallel to the interest given by the society to these new technologies. Graph theory, algebraic graph theory and other mathematical tools have been developed to tackle in rigorous and systematic ways problems of interest for the increasingly connected society. Thanks to the abstraction given by mathematics, it became possible to discover how some well-known and studied processes and phenomena find their natural and most convenient description (from an engineering point of view) precisely in the framework of networked systems and graph theory.
A useful class of probabilistic dynamics on graph is represented by Markov Chains. This stochastic model is used in many computational algorithms to sample from a probability distribution that is not directly available nor fully known. This is done by appropriately designing a stochastic evolution that converges (mixes) to an equilibrium coinciding with the target distribution.
In order to accelerate the convergence speed of probabilistic algorithms, other stochastic processes have been proposed such as Lifted Markov Chains, where a non-Markovian effect that can be seen as added local memory is introduced by considering Markov evaluations on an enlarged state space. The latter is based on the concept of graph lifting, and the induced memory effects can be translated into connectivity patterns among the nodes of the extended network.
In this thesis we will borrow (from the above mentioned probabilistic models) the idea of lift on a graph and how lift can be designed to add a local memory to processes naturally described on networks. We will then apply these ideas to a completely different context: controlled networks of interconnected systems.
Solving control problem for large scale networks is particularly difficult; for many complex networks the system parameters are not precisely known and the connections between different systems are several and intricate. Among all the possible aspects of the control problem for complex networks we will mainly constrain our attention to the effort necessary to control the system (independently on the specific input applied to it), which is practically and conveniently quantified by some metrics related to the so-called control energy. Control energy typically grows exponentially with the number of components (nodes) in the network if the number of control inputs remains constant.
Each node in the network is connected to several other units, hence it may receive control inputs from multiple directions. The main idea we will develop in the thesis is to provide nodes with a local memory that keeps track of the direction from which the inputs are received and with a computational power that allows them to treat and spread the received inputs in the most efficient way through the network.
The tool we will exploit to describe these extra capabilities given to the nodes is exactly the lift of graph mentioned above.
We will assess how, by properly adding memory and computational power to nodes, the energy required to control the system grows more slowly as the quantity of vertices in the network (and hence its dimension) increases and the number of external control inputs remains constant.

To make the analysis meaningful we will impose some constraints on the allowed lifted dynamics. Thanks to the extra capabilities provided, each node is allowed to spread signals through the network differently from the original system, however it cannot inject additional energy by amplifying signals. The imposed constraints play a crucial role in assessing whether the advantages given by changing the internal node dynamics effectively stem from memory effects.

The thesis is structured as follows. After a brief summary of the background necessary to understand the treated topics (chapter I) we will describe how it is possible to lift a controlled network (chapter II). Subsequently, a comparison will be performed between the control energy performances for a given network and its lifted counterpart. Most notably, this analysis will be conducted for a couple of well-known and widely used network topologies: the line (Chapter III) and the grid (Chapter IV). We will prove analytically and numerically how for these network topologies, by appropriately performing lift, the expanded network with extra memory and computation capabilities performs better than the simple counterpart.

Chapter 1

Preliminaries

In this chapter we introduce the background necessary to understand the main contributions of this thesis. In particular, we will review some of the result available in the literature about structural and energy-related controllability by introducing the main tools exploited in the analysis. We will then explain what it means to perform a lift on a graph and apply it to the context of Lifted Markov Chains to show the usefulness of this operation.

1.1 Controllability of Linear Time-invariant Systems

One of the main questions that arises when we analyze a system is what are the states that can be reached and the trajectories that can be followed by suitably choosing the available control inputs. In this work we will consider only linear time-invariant (LTI) and discrete-time systems which are often described in state space form, i.e., by a system of equations of the form:

$$\Sigma : \begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{1.1}$$

where:

- $x(t) \in \mathbb{R}^n$ is the vector describing the state of the system;
- $u(t) \in \mathbb{R}^m$ is the input vector;
- $y(t) \in \mathbb{R}^p$ is the output vector which describes how the measurement taken by sensors are influenced by the state of the system and the input;
- $A \in \mathbb{R}^{n\times n}$ is the state matrix which describes how the system behaves when no input is applied;
- $B \in \mathbb{R}^{n\times m}$ is the input to state matrix which describes how the input act on the system;
- $C \in \mathbb{R}^{p\times n}$ is the output matrix and describes the influence of the actual state on the output vector;
- $D \in \mathbb{R}^{p\times m}$ is the feedforward matrix that describes how the output of the system is influenced by the actual input.

After introducing the state space model for a linear time-invariant discrete-time system we now introduce in detail the controllability problem.

1.1.1 Classical controllability

Before even considering the problem of designing an input to drive the system to some target state it is necessary to check if the desired configuration can be reached.

Let Σ be a linear time-invariant discrete-time system. If the initial state is zero, $x(0) = 0$, and the applied input is

$$u(0), u(1), \dots, u(k-1) \tag{1.2}$$

the state that is reached at time k is

$$x(k) = \sum_{\sigma=0}^{k-1} A^{k-1-\sigma} B u(\sigma) = \begin{bmatrix} B & AB & \dots & A^{k-1}B \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(0) \end{bmatrix} \tag{1.3}$$

The set X_k^R of reachable states in k steps is the image, namely, the set of all possible linear combinations of the columns, of the matrix

$$\mathcal{C}_k = \begin{bmatrix} B & AB & \dots & A^{k-1}B \end{bmatrix} \tag{1.4}$$

The property of a system to be steered to any desired state via suitable inputs is known in systems theory as **controllability**, the formal definition is given below.

Definition 1 (Controllability) *A system is said to be controllable (reachable) in k steps if and only if for every state $x_f \in \mathbb{R}^n$ there exists an input sequence such that $x(k) = x_f$ with $x(0) = 0$.*

There exist different controllability conditions which allow us to check if an LTI system is controllable from a given set of inputs. A well-known condition is the **Kalman's rank condition** which makes use of the **controllability matrix**, defined as:

$$\mathcal{C} := \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \tag{1.5}$$

Theorem 1 (Kalman's rank condition) *The LTI system (A, B) is controllable if and only if the controllability matrix $\mathcal{C}$ has full row rank.*

If the matrix $\mathcal{C}_k$ in (1.4) has full row rank, then the system is said to be controllable in k steps. Another interesting controllability criterion is the so called **Popov, Belevitch and Hautus (PBH) criterion** and its corollaries. Employing the PBH criterion, the controllability analysis of an n-dimensional LTI system is done by studying the rank of the corresponding PBH matrix, namely:

$$\begin{bmatrix} A - zI_n & B \end{bmatrix} \tag{1.6}$$

as z varies over the complex field.

Theorem 2 (PBH test) *An n-dimensional discrete-time LTI system Σ as in* (1.1) *is controllable if and only if the matrix* (1.6) *has full row rank for all $z \in \mathbb{C}$.*

Remark 1 *For arbitrary matrices A and B, the matrix* (1.6) *has row full rank, equal to n, for any z that is not in the spectrum of A. Therefore, the only values $z \in \mathbb{C}$ for which it can have rank less than n are the eigenvalues of the matrix A.*

We will now briefly describe how it is possible to design an input to reach the target state in a finite time horizon $[0, k]$. It is easy to see that there exists a sequence $u(0), u(1), \dots, u(k-1)$ that steers the system from the initial state $x(0) = x_0$ to the final state $x(k) = x_f$ if and only if

$$x_f - A^k x_0 \in \operatorname{Im} \begin{bmatrix} B & AB & \dots & A^{k-1}B \end{bmatrix} = X_R^k, \tag{1.7}$$

where $\operatorname{Im}(\cdot)$ denotes the image of a matrix, namely if and only if $x_f - A^k x_0$ is reachable, from the zero state, in k steps. Therefore, in order to determine an input sequence we must solve the equation

$$x_f - A^k x_0 = \mathcal{C}_k u \tag{1.8}$$

where u is the vector:

$$u = \begin{bmatrix} u(k-1)^T & u(k-2)^T & \dots & u(0)^T \end{bmatrix}^T \in \mathbb{R}^{km} \tag{1.9}$$

we consider as norm of the vector **u** the norm induced by the standard inner product in the input space:

$$E(u,k) = ||u||_2 = \langle u, u \rangle = \left(\sum_{\sigma=0}^{k-1} u^T(\sigma) u(\sigma) \right)^{\frac{1}{2}} \tag{1.10}$$

This norm is often referred as **control energy** of the input signal u.

We need now to introduce the so called (controllability) Gramian of the system which will be one of the core quantities to be considered in the energy-related controllability analysis that we will describe later on. Given the system Σ described by equation (1.1), the **Controllability Gramian** in k steps is defined as:

$$\mathcal{W}_k := \mathcal{C}_k \mathcal{C}_k^T = \sum_{\sigma=0}^{k-1} A^{k-1-\sigma} B B^T (A^{k-1-\sigma})^T \tag{1.11}$$

The Gramian $\mathcal{W}_k$ is always positive semidefinite and it is positive definite if and only if the system is controllable.
Using the controllability gramian it is possible to determine the minimum-energy control input $\bar{\mathbf{u}}$ that drives the system from the initial state $x(0) = x_0$ to the final state $x(k) = x_f$. Such input is defined as

$$\bar{u} = \mathcal{C}_k^T \eta, \tag{1.12}$$

where $\eta \in \mathbb{R}^n$ is the solution of the auxiliary equation

$$\mathcal{W}_k \eta = x_f - A^k x_0. \tag{1.13}$$

For sake of completeness we show below that the input $\bar{u}$ is the minimum energy one.

Proposition 1 *The solution $\bar{u}$ obtained through* (1.12) *has minimum energy among all the solutions to* (1.8).

Proof: The solution $\bar{u} = \mathcal{C}_k^T \eta$ belongs to $\operatorname{Im} \mathcal{C}_k^T = (\ker \mathcal{C}_k)^\perp$. Any other solution can be expressed through $\bar{u} + v$, where v is the solution to $\mathcal{C}_k v = 0$ and hence a generic element of $\ker \mathcal{C}_k$. Since $\bar{u}$ and v are orthogonal, it holds true that $||\bar{u} + v||^2 = ||\bar{u}||^2 + ||v||^2$, which shows that $\bar{u}$ is the minimum norm solution to (1.8). ■
Another useful controllability definition is the so-called **output controllability**, inspired by the fact that in many engineering applications we are interested in controlling the output of the system rather than its full state.

Definition 2 (Output controllability) *The LTI system* (1.1) *is said to be output controllable in k steps if and only if for every output $y_f \in \mathbb{R}^n$ there exists an input sequence such that $y(k) = y_f$ with $y(0) = 0$.*

Output controllability can be checked similarly as standard controllability by applying the Kalman's rank condition to the so-called **output controllability matrix** defined as:

$$\mathcal{C}^O := C \begin{bmatrix} B & AB & \cdots & AB^{n-1} \end{bmatrix} = \begin{bmatrix} CB & CAB & \cdots & CAB^{n-1} \end{bmatrix} \tag{1.14}$$

The **output controllability Gramian** is then given by:

$$\mathcal{W}_k^O = C \mathcal{W}_k C^T \tag{1.15}$$

The minimum energy input steering the system output to the final state y_f can be determined with a procedure similar to the one illustrated before.

Finally one may be interested in finding the minimum energy input signal controlling the system to a certain final state over an infinite horizon. In this case we need to introduce the infinite horizon (output) controllability Gramian which is defined as:

$$\mathcal{W} = \sum_{k=0}^{\infty} A^k B B^T (A^k)^T. \tag{1.16}$$

Note that $\mathcal{W}$ is well-defined only if the matrix A is Schur stable, i.e., all its eigenvalues have modulus smaller than 1. In this case the Gramian can be computed as the solution of the discrete-time Lyapunov equation:

$$\mathcal{W} = A\mathcal{W}A^T + BB^T \tag{1.17}$$

The controllability definitions described up to now and the related Kalman and PBH tests are very simple and effective. Indeed, thanks to the aforementioned theory it is possible to check whether there exist control inputs able to steer the system to a desired target state and, if so, to design them. However as the dimension of the system grows some limitations of this theory arise.

1.2 Controllability on networks

In the last decade the interest of research in control theory has moved from the development of algorithms and mathematical tools to study and govern single actuators to the control of increasingly interconnected networks of heterogeneous components. This interest is motivated by the vast number of applications which can be framed in the context of networks such as power networks, social networks, communication between groups of robots and traffic flow networks. In these systems complexity arises primarily from the usually large and intricate interconnection patterns and the rich dynamics of the isolated units. Due to the increasing complexity of networks the focus has shifted toward the framework of **large-scale networks**.

In this dissertation we will consider only discrete-time, linear and time-invariant networks which can be described in the state space form (1.1) as explained below.

A **network** can be represented by a **directed graph** $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} := \{1, 2, ...n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges.

We define the **adjacency matrix** of $\mathcal{G}$ as $A = [a_{ij}]$ where $a_{ij} \in \mathbb{R}$ are the weight associated to the edge $(j, i) \in \mathcal{E}$, moreover $a_{ij} = 0$ if $(j, i) \notin \mathcal{E}$.

Example 1 (Adjacency matrix and related graph)
Consider the adjacency matrix:

$$A = \begin{bmatrix} 0 & 0 & 0.1 \\ 0.2 & 0 & 0.3 \\ 0 & 0.4 & 0 \end{bmatrix} \tag{1.18}$$

The related graph is shown in figure 1.1.

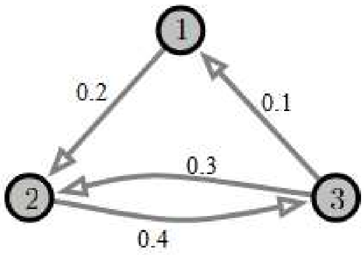

Figure 1.1: First example of graph

The state of the system is described by the vector $x(t) \in \mathbb{R}^n$, whose entry $x_i(t)$ represents the state of node $i \in \mathcal{V}$ at time $t \in \mathbb{N}_{\geq 0}$. The evolution of the system is given by the difference equation:

$$x(t+1) = Ax(t) \tag{1.19}$$

Hence the edge $(i, j) \in \mathcal{E}$ and the associated weight, which are described by the quantity $[A]_{ji}$, tell us how the information on the state of node i at time t influences the state of node j at time $t+1$.

Until now the system has been considered without any input. One of the main tasks that needs to be handled when dealing with large-scale systems is the definition of the set of nodes that can be controlled by means of an external input. This choice affects the controllability properties of the system.

We assume the set of control nodes to be given and to be equal to $\mathcal{K} = \{k_1, ..., k_m\} \subseteq \mathcal{V}$. Let $B_{\mathcal{K}}$ be the input matrix, where we made explicit the dependence on the choice of input nodes. If we define the input signal $u_{\mathcal{K}}(t) \in \mathbb{R}^m$, the difference equation describing the evolution of the network state becomes:

$$x(t+1) = Ax(t) + B_{\mathcal{K}} u_{\mathcal{K}}(t). \tag{1.20}$$

With this description we have found the state equation as in (1.1) for the network system, hence controllability can be checked by applying Kalman's rank condition or PBH test as before.

The problem of controlling large-scale networks can be tackled by different points of view; beyond the classical definition that can be found in all the books of system theory, the concepts of **energy-related controllability** (or practical controllability) and **structural controllability** have been developed.

Energy-related controllability focuses on the more concrete assumption that even though a system is controllable by a theoretical perspective, the effort necessary to reach a desired state is not always sustainable in practice. The main aspects that influence the energy necessary to steer the system to the target state are the topology of the network (i.e., how the single units composing the system are interconnected), the choice of the control nodes, and the dimension of the network.

Structural controllability theory gives necessary and sufficient conditions for almost every network system sharing the same connectivity pattern to be controllable with the usual meaning.

In the next two subsections we will describe more in detail these controllability concepts.

1.2.1 Energy-related controllability

The classical notion of controllability is qualitative, in that it does not quantify the difficulty of the control task. This effort can be measured via the **control energy** defined in (1.10). Let $\bar{u}_{\mathcal{K}}(\cdot)$ be the minimum energy input defined in (1.12) where we made explicit the dependence on the control nodes, and let $\lambda_{\min}(\mathcal{W}_{\mathcal{K},T})$ denote the smallest eigenvalue of the controllability Gramian of the system $\mathcal{W}_{\mathcal{K},T}$. It is possible to notice that:

$$E(\bar{u}_{\mathcal{K}}, T) = \sum_{\tau=0}^{T-1} \|\bar{u}_{\mathcal{K}}(\tau)\|_2^2 = x_f^T \mathcal{W}_{\mathcal{K},T}^{-1} x_f \leq \lambda_{\min}^{-1}(\mathcal{W}_{\mathcal{K},T}) \|x_f\|_2^2 \tag{1.21}$$

We can therefore quantify the effort required to control the network in terms of minimum-energy control inputs.

Different metrics for the overall required energy given any target final state have been proposed over the years:

1. $\lambda_{\min}(\mathcal{W}_{\mathcal{K},T})$, which, in view of (1.21), is related to the worst-case control energy. The system is less controllable as this metric decreases.

2. $\text{trace}(\mathcal{W}_{\mathcal{K},T}^{-1})$ measures the average control energy over unit-norm random target states. The system is less controllable as this metric increases.

3. $\det(\mathcal{W}_{\mathcal{K},T})$ is proportional to the volume of the ellipsoid containing the states that can be reached with unit control energy.

4. $\text{trace}(\mathcal{W}_{\mathcal{K},T})$ is a pseudo-metric that is often used in literature as a measure of controllability. Note that the system can be uncontrollable even if this metric takes large values.

All these metrics depends on the controllability Gramian and not on the specific target state, hence they allow us to quantify the difficulty of the control task independently on the final state we want to reach. If A is symmetric (or close to be symmetric) and the number of control nodes remain constant as the network size increases, the control effort, which can be quantified using the metrics above, grows exponentially with the size of the network.

A good introduction to energy related controllability and how control energy is influenced by the number of control nodes can be found in [1].

1.2.2 Structural Controllability

The last definition of controllability we will consider, even if only briefly, is that of structural controllability. The idea behind structural controllability is that for many complex networks the system parameters are not precisely known, in particular the weights of the links can not be measured effectively, we only know when there is an edge between two nodes or not. Therefore for systems over networks it is hard to verify numerically Kalman's rank condition. Systems showing these problems are called structured systems, the formal definition is given below.

Definition 3 (Structured system) *An LTI system $(A, B_{\mathcal{K}})$ is a structured system if the elements in A and $B_{\mathcal{K}}$ are either fixed zeros or independent free parameters. The corresponding matrices A and $B_{\mathcal{K}}$ are called structured matrices.*

We are now ready to give the definition of structural controllability.

Definition 4 (Structural controllability) *The system $(A, B_{\mathcal{K}})$ is structurally controllable if we can set the nonzero elements in A and $B_{\mathcal{K}}$ such that the resulting system is controllable in the usual sense.*

The power of structural controllability comes from the fact that if we are able to prove that a system is structurally controllable then it is controllable for almost all possible parameters realizations.

In the following, after giving a couple of required definition, we will introduce a criterion useful to check when the systems is structurally controllable.
Consider an LTI system $(A, B_{\mathcal{K}})$ represented by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{K} \subset \mathcal{V}$ is the set of control (input) vertices.

Definition 5 *A vertex x_i is inaccessible if there are no directed paths reaching x_i from the input vertices.*

Let $\mathcal{S} \subset \mathcal{V} \setminus \mathcal{K}$ and $T(\mathcal{S})$ be the in-neighborhood of $\mathcal{S}$, i.e the set of all the vertices v_j for which there exist a directed edge from v_j to some other node in $\mathcal{S}$.

Definition 6 *The directed graph $\mathcal{G}$ contains a* ***dilation*** *if there is a subset of nodes $\mathcal{S} \subset \mathcal{V} \setminus \mathcal{K}$ such that $T(\mathcal{S})$ has fewer nodes than $\mathcal{S}$ itself.*

The above mentioned criterion is now stated as a theorem.
Theorem 3 (Lin, 1974) *The system $(A,B_{\mathcal{K}})$ is not structurally controllable if and only if it has inaccessible nodes or dilations.*

In figure 1.2 it is depicted an example of what we mean with inaccessible nodes and dilations. The nodes x_1 and x_2 on the left picture are inaccessible from the control node x_6. Variations in the input u_1 can not influence these two nodes since there does not exist any path joining the input u_1 with any of x_1, x_2. On the right picture instead we can appreciate a dilation, the node set $\mathcal{S} = \{x_3, x_4\}$ has as neighboring set $T(\mathcal{S}) = \{x_5\}$, which contains only one node, hence the size of $T(\mathcal{S})$ is smaller than $\mathcal{S}$ implying that a single node in $T(\mathcal{S})$ is requested to control two nodes in $\mathcal{S}$.

While more sophisticated and systematic ways to check structural controllability were introduced by Lin, they go beyond the scope of the present work. One relevant paper which gives a complete and effective introduction to structural controllability is [2].

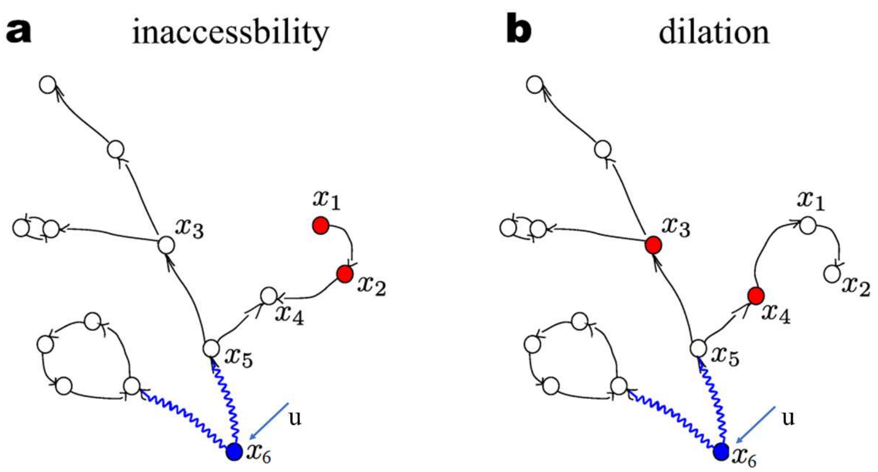

Figure 1.2: Inaccessible nodes and dilations.

1.3 Lift of Markov Chains

In this section we introduce Markov Chains on graphs and Lifted Markov Chains. The latter have been mainly considered as a useful technique to speed up some performances of a classical

Markov chain, such as mixing times, by effectively introducing memory in the dynamics. The scope of this preliminary section is only to give a brief overview to these concepts which inspired the ideas behind the thesis work. Starting from next chapter we will borrow from this framework only the idea of lift on graphs.

1.3.1 Markov chains

The significance of algorithms based on Markov chains has been extensively validated. In the realm of computer science, random walks and Markov chain Monte Carlo play a crucial role in numerous randomized algorithms. Practical applications of Markov chains encompass the Google page ranking algorithm, solving average consensus problems, employing Markov chain Monte Carlo methods to sample from unknown target distributions, and implementing simulated annealing techniques. We now introduce some fundamental definitions that are essential to grasp the concept of Markov chains and how they are employed in practical algorithms to engineer a stochastic evolution that converges (mixes) to a target distribution.

Definition 7 *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A family of random variables $\{X(t); t \in T\}$, $T \subseteq Z$ defined on Ω is called a discrete-time stochastic process.*

A stochastic process is called a Markov process if it satisfies the so-called Markovian property which can be described as

"The future of the process depends on the past only through the present"

The formal definition is given below:
Definition 8 *Consider a family of random variables $X = \{X(t); t \in \mathbb{N}\}$ all taking values in the finite or countable set $\mathcal{X}$ of cardinality n. The process is called a Markov chain if it satisfies the Markov property:*

$$\mathbb{P}(X(t+1) = x | X(0) = x_0, X(1) = x_1, \dots, X(t) = x_t) = \mathbb{P}(X(t+1) = x | X(t) = x_t) \tag{1.22}$$

$\forall (x_0, x_1, \dots, x_t, x) \in \mathcal{X}^{t+2}, \forall t \geq 0$
Definition 9 *The **distribution** of X(t) is defined as:*

$$\pi_i(t) = \mathbb{P}(X(t) = i) \tag{1.23}$$

Definition 10 *Let $i, j \in \mathcal{X}$ The **one-step transition probability** is the probability of $X(t+1)$ being in state j given $X(t)$ is in state i:*

$$p_{ij}(t) = \mathbb{P}(X(t+1) = j | X(t) = i) \tag{1.24}$$

In many application the transition probabilities does not depend on time, in this case the Markov chain is called time-homogeneous.
From now on we will consider only time-homogeneous Markov Chains, the probability of $X(t+1)$ being in state j is given by

$$\pi_j(t+1) = \sum_i p_{ij} \pi_i(t) \tag{1.25}$$

The transition probabilities can be arranged in a matrix P, and the distribution of every state at time t in a vector $\pi(t)$, i.e.,

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \qquad \pi(t)^T = \begin{bmatrix} \pi_1(t) & \dots & \pi_n(t) \end{bmatrix}^T \tag{1.26}$$

The matrix P enjoys the following properties:

1. $p_{ij} \geq 0 \; \forall i, \forall j$
2. $\sum_j p_{ij} = 1 \; \forall i$

A matrix satisfying this two properties is called **stochastic**. The overall distribution on the node set at time t can be easily expressed thanks to this notation as:

$$\pi(t+1) = P^T \pi(t) \tag{1.27}$$

Definition 11 *A distribution $\tilde{\pi}$ is called stationary for the Markov chain X with transitional probability matrix P if it satisfies*

$$\tilde{\pi} = P^T \tilde{\pi} \tag{1.28}$$

Definition 12 *A chain $\mathcal{X}$ is said to be irreducible (or ergodic) if its transition probability matrix P is irreducible, i.e.*

$$\sum_{i=0}^{n-1} P^i > 0$$

Definition 13 *A chain with stationary distribution π is said to be reversible if:*

$$\tilde{\pi}_i p_{ij} = \tilde{\pi}_j p_{ji} \tag{1.29}$$

Markov chains are often represented on graphs.
Consider a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ where $\mathcal{E}$ is the set of edges and $\mathcal{X}$ is the same countable set of n elements of before, now $\mathcal{X}$ is considered to be the set of nodes which we will label as $\mathcal{X} = \{i = 1 \ldots N\}$. The transpose of the transition probability matrix P is now thought as the adjacency matrix of the graph, i.e., its entries are the weights assigned to the edges connecting the nodes in $\mathcal{X}$. An important constraint on the matrix P is that $[P]_{ij} = 0 \; \forall i, j$ s.t $(i, j) \notin \mathcal{E}$. An example of how a Markov chain can be described on graph is shown in figure 1.3.

One of the task that can be performed using Markov chains over graphs is that of designing a mixing dynamics, the problem is formally defined below.
Problem (Design of mixing dynamics): Design a discrete-time stochastic map that converges toward a target distribution $\tilde{\pi}$ on $\mathcal{X}$ while respecting the locality associated to the graph.
A common approach to solve the mixing problem is to converge toward the steady state distribution $\tilde{\pi} \in \mathbb{P}$ by iterating the linear stochastic discrete time map described by equation (1.27).
Observation: P must be irreducible to allow $\pi(t)$ to converge to π for any $\pi(0)$.

This approach is often referred in literature as **random walk** as we can imagine a pawn following a certain path among all possible nodes of the graph. The pawn has a certain probability to move in a direction and this probability is described by the entries of the matrix P. At the end of the transition the presence of the pawn in one of vertices of the graph is given by a probability distribution.

We now analyze an example of how a Markov chains can be designed in order to converge toward a uniform distribution over states in $\mathcal{X}$.

Example 2 (Markov chain for uniform mixing)
We aim to sample from a target distribution $\tilde{\pi}$ on a finite set $\mathcal{X}$ by running an irreducible Markov chain on $\mathcal{X}$ with transition probability matrix P built so that $\tilde{\pi}$ is the stationary distribution. We consider the set $\mathcal{X} = \{1, 2, \ldots, N\}$ and the target distribution $\tilde{\pi} = \frac{1}{n}\mathbb{1}_n$.
We set:

$$\begin{aligned} p_{ij} = p_{ji} &= \frac{1}{2} \; \textit{where} \; j = i \pm 1 \;\; i, j \in \mathcal{X} \\ p_{11} = p_{nn} &= \frac{1}{2} \end{aligned} \tag{1.30}$$

The underlying graph is depicted below.

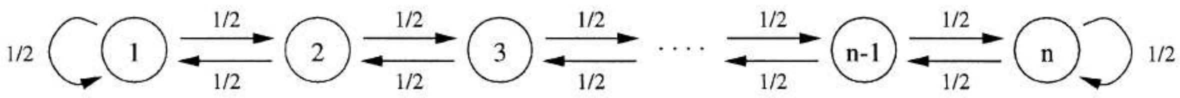

Figure 1.3: Example of Markov Chain

This chain can be proved to converge to the desired distribution over element of $\mathcal{X}$ (it derives from the fact that the transition probability matrix is doubly stochastic and primitive).
The convergence can be proven to take order N^2 steps.

1.3.2 Lifted Markov Chains

Lifted Markov chains (LMC) are a stochastic evolution which can be used to accelerate convergence with respect to classical Markov chains (described in the previous section) to a target distribution. Considering random walks, the effect of the lift is that of "adding a local memory" to the walker which keeps partial track of the previous nodes explored. The lifted Markov chain can be obtained from a classical Markov chain by performing a certain operation on the underlying graph of the MC.

We consider again the setting of the previous section, i.e., a classical Markov chain on a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$.
The lifted Markov chain can be obtained starting from $\mathcal{G}$ as formally described below.
Definition 14 (Lifted graph) *A graph $\hat{\mathcal{G}} = (\hat{\mathcal{X}}, \hat{\mathcal{E}})$ on $\hat{N}$ nodes is called a lift of $\mathcal{G}$ if there exist a surjective map $\zeta : \hat{\mathcal{X}} \to \mathcal{X}$ such that:*

$$(i, j) \in \hat{\mathcal{E}} \implies (\zeta(i), \zeta(j)) \in \mathcal{E}. \tag{1.31}$$

We denote ζ^{-1} the map that takes as input a single node $k \in \mathcal{X}$ and outputs the set of nodes $j \in \hat{\mathcal{V}}$ for which $\zeta(j) = k$.
Intuitively, we may think of lifting as the action of adding a certain number of copies of each node $j \in \mathcal{X}$ in the vertex set $\hat{\mathcal{X}}$.
The lift is said to be **regular** if the set $\hat{\mathcal{X}}_j = \{k \,|\, \zeta^{-1}(k) = j\}$ has the same cardinality $\forall j \in \mathcal{X}$.
In a regular lifted graph we associate to each element in $\mathcal{X}$ the same number of elements in $\hat{\mathcal{X}}$.
This concept of regular lift will be extensively used in the next chapters. Example 3 clarifies how a regular lift can be obtained from a simple graph.

Example 3 (Regular Lift)
Consider the graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ where $\mathcal{X} = \{1, 2, 3\}$ and $\mathcal{E} = \{(1, 2), (2, 3), (3, 1)\}$.
A regular lift of $\mathcal{G}$ is $\hat{\mathcal{G}} = (\hat{\mathcal{X}}, \hat{\mathcal{E}})$ where $\hat{\mathcal{X}}$=$\{1, 2, 3, 4, 5, 6\}$, we define the map $\zeta(k) = ((k - 1)$ mod $3) + 1$, the map ζ^{-1} is such that $\zeta^{-1}(i) = \{i, i + 3\}$.
A possible edge set is given by $\hat{\mathcal{E}} = \{(1, 2), (2, 3), (3, 1), (4, 5), (5, 6), (6, 4)\}$.

Let x be a probability distribution over the lifted graph nodes which belongs to the set $\hat{\mathcal{X}}$. The lifted Markov chains dynamics is then obtained by iterating a stochastic (Markov) evolution on the lifted graph:

$$x(t + 1) = Ax(t) \tag{1.32}$$

The matrix A may be thought as the adjacency matrix of $\hat{\mathcal{G}}$, and it must satisfies the locality constraints on the graph $\hat{\mathcal{G}}$ induced by the underlying $\mathcal{G}$, i.e., $A_{j,l} \neq 0$ only if $(\zeta(j), \zeta(l))$ is an edge of $\mathcal{G}$.

If we are however interested in the distribution $\pi(t)$ on $\mathcal{X}$ at time t, this can be obtained as the marginal of $x(t)$, defined as $\pi_k = \sum_{j\in\zeta^{-1}(k)} x_j$.
The relation on the marginal can be hence represented in vector form as:

$$\pi = Cx \tag{1.33}$$

where the entries of C can be either 0 or 1.

When considering two distinct approaches (e.g., Markov Chains vs. Lifted Markov Chains), it is crucial to comprehend their dissimilarities and identify the specific constraints under which it becomes meaningful to make a comparison between their performances. Consequently, some constraint needs to be imposed on the lift evolution, and the improvement in mixing performance critically depend on them. A complete description of mixing performances results can be found in [3]. The full knowledge of how constraints impact mixing performances goes beyond the scopes of this introductory section, hence, in the following we list only some among the ones discussed in the above paper to highlight their importance and the possible differences between LMC and MC.

1. **Constraints on the initialization of the lift**:
   One must specify how to choose the initial state of the stochastic dynamic, we distinguish two cases:

   - it is possible to initialize the lifted dynamics by choosing the initial distribution on the nodes and how to lift $\pi_k(0)$ onto a distribution over its lifted nodes. The designed initialization has to be a linear map $F : \pi(0) \to x(0)$ such that:

   $$CF\pi(0) = \pi(0) \quad \forall \pi(0) \tag{1.34}$$

   - there is no control on the initialization of the lift dynamics. The set of relevant initial condition is the whole $\mathbb{P}_{\hat{N}}$.

2. **Invariance of the target marginal** For a Markov chain mixing is toward its invariant distribution, hence $P\tilde{\pi} = \tilde{\pi}$. For a lifted Markov chain, however, if the marginal converges to $\tilde{\pi}$, in the transient $Cx(t) = \tilde{\pi}$ does not necessary imply that $Cx(t+1) = \tilde{\pi}$. We thus identify two cases:

   - it is imposed $Cx(t) = \tilde{\pi}\ \forall t > 0$ whenever $Cx(0) = \tilde{\pi}$ for all the $x(0)$ belonging to the set of initial conditions.
   - it is allowed $Cx(t) \neq \tilde{\pi}$ for some $t \geq 0$ and some $x(0)$ even when $Cx(0) = \tilde{\pi}$.

3. **Reducibility of the lift** A Markov chain P that globally converges to a unique stationary distribution $\tilde{\pi}$ with $\pi_i > 0$, $\forall i$, must be irreducible, however the same does not necessary apply to the lifted Markov Chain described by A.
   We discuss two cases:

   - the lifted Markov chain A is allowed to be reducible.
   - the lifted chain A must be irreducible.

We give now an example of lifted Markov chain that will be of paramount importance for the rest of thesis work.

Example 4 (The Diaconis Lift)
We consider the same Markov chain described in example 1 where for any given node one moves with equal probability to either of the two neighbors, we recall that this chain is reversible.
The idea now is to build a non-reversible Markov Chain where we make the walker next step depend on its last move. The lift is build by associating to each node $k \in \mathcal{X}$ of the original graph two nodes $(\pm, k)$ which indicates if the current position k has been reached from k+1 or k-1 respectively.
The lifted node set is $\hat{\mathcal{X}} = \{(s,k) : k = 1,2,\dots,N \text{ and } s \in \pm 1\}$. The allowed edge set consists of the edges $\hat{\mathcal{E}} = \{((s', k \pm 1 \mod N), (s,k)) : k = 1,2,\dots,N \text{ and } s, s' \in \pm 1\}$.
By exploiting the Kronecker product (described in appendix A.1) the lifted transition map can be written as:

$$A = \sum_{i,j \in \pm 1} Q_{i,j} e_i e_j^\top \otimes P^{(j)} \tag{1.35}$$

where $e_{\pm 1}$ are the columns vectors $(1,0)^T$ and $(0,1)^T$, $P^{(\pm 1)}$ is the clockwise and anticlockwise rotation on the $\mathcal{G}$. Q is a stochastic matrix whose entries are defined as:

$$Q_{+1,+1} = Q_{-1,-1} = 1 - 1/N \text{ and } Q_{+1,-1} = Q_{-1,+1} = 1/N. \tag{1.36}$$

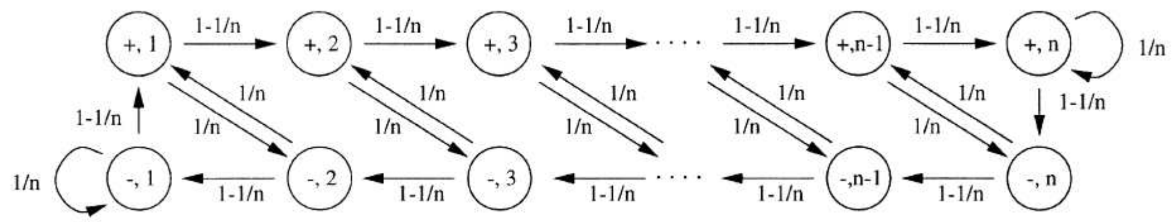

Figure 1.4: Diaconis Lift

Thanks to this lift it is possible to achieve mixing in order N step (as formally proved in [4]), which is really less than order N^2 steps required by the corresponding Markov chain of example 2.

The mixing problem described in this section is a stabilization problem, in fact we would like to study whether the system converges to the target distribution on the nodes of the graph. However controllability described in the previous section concerns the capability of steering the system to a desired state through an appropriate input signal, this problem can not be treated in the context of Markov chains, as no input is employed. One can hence wonder which are the links between the topics presented in this preliminary section.

Although we have described how the lift operation can be employed in the particular contest of MC and LMC to achieve improvement in mixing performances this is a completely general operation and can be applied to any graph. In the next chapter we will describe how to perform lift on controlled network of homogeneous component, and in chapters 3 and 4 we prove both analytically and through some simulations the advantages in terms of controllability we have by using lifts.

Chapter 2

Lifts of Controlled Networks

In this chapter we borrow the idea of lift of graphs which was previously described and used in the context of Lifted Markov Chains to explore whether it is possible to exploit it to improve controllability performances of a general network fed with a certain number of inputs.
In particular we are interested in proving the advantages of this construction as the cardinality of the network grows and the number of control nodes remains constant.
The idea is that thanks to the local memory add to the graph by the lift operation we will be able to improve the "directionality" of the control input and reach easily the nodes which are farther away from the point where the input is applied. We will describe a way to design an homogeneous local lift for a network with certain connectivity patterns.

2.1 The underlying network

In this section we present a method to describe the underlying non-lifted network.
We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, ..., n\}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Moreover we label with $\mathcal{K}$ the set of control nodes. The state of the network at time $t \in \mathbb{N}$ is described by the vector $x(t) = \begin{bmatrix} x_1(t) & \dots x_n(t) \end{bmatrix}$, $x(t) \in \mathbb{R}^{nN}$ whose entry $x_i(t) \in \mathbb{R}^N$ is the state of node $i \in \mathcal{V}$.

We may think to the network as composed by n agents each one of them with its internal dynamics. The dynamics of the i-th agent can be described by the linear time-invariant discrete-time system:

$$\Sigma_i : \begin{cases} x_i(t+1) = A_i x_i(t) + B_{v,i} v_i(t) + B_{u,i} u_i(t) \\ y_i(t) = C_i x_i(t) \end{cases} \tag{2.1}$$

Where

- $A_i \in \mathbb{R}^{N \times N}$ is the state matrix of the node;
- $B_{v,i} \in \mathbb{R}^{N \times q}$ is the input matrix describing how the input $v_i(t) \in \mathbb{R}^q$ from other nodes in the networks enter the state of node i.
- $B_{u,i} \in \mathbb{R}^{N \times 1}$ is the input matrix which describes how the input $u_i(t) \in \mathbb{R}$, from the external environment enters node i.
- $C_i \in \mathbb{R}^{p \times N}$ is the output matrix which describes how the state of the node influences its output $y_i \in \mathbb{R}^p$.

For sake of simplicity we consider all the agents of the network to be identical, with the dynamics described by the same matrices $A_i, B_{v,i}, B_{u,i}, C_i$. From now on these matrices are hence considered to be independent on the specific agent i. We maintain the subscript i to highlight that these matrices are associated to the evolution of a single agent. Thanks to this assumption it is possible to describe the overall system as a network of diffusively coupled linear systems by adapting to the discrete time case the framework introduced in [5] chapter 8.

Exploiting the Kronecker product (introduced in appendix A.1) we are now able to connect the different agents and describe the overall network system in an easy and compact way:

$$\Sigma : \begin{cases} x(t+1) = (I_n \otimes A_i)x(t) + (I_n \otimes B_{v,i})v(t) + B_u u(t) \\ y(t) = (I_n \otimes C_i)x(t) \end{cases} \tag{2.2}$$

Where:

$$B_u = \tilde{B}_u \otimes B_{u,i} \qquad B_u \in \mathbb{R}^{n\times m}, \qquad v(t) = Ly(t). \tag{2.3}$$

The matrix $L \in \mathbb{R}^{qn\times np}$ describes the connections between the different nodes of the network, its entries are either 0 or 1. In particular let $y_{c,j}$ be the j-th output from node $c \in \mathcal{V}$ and $v_{d,s}$ the s-th output from node $d \in \mathcal{V}$, $[L]_{(q-1)d+s,(p-1)c+j} = 1$ iff $y_{c,j}$ is connected to $v_{d,s}$.
The matrix $\tilde{B}_u \in \mathbb{R}^{n\times m}$ maps an input to a specific control node, hence its columns are canonical vectors.
The topology of the network is completely described by the two matrices L an $\tilde{B}_u$.
The overall state equation can be written exploiting the Kronecker product as:

$$x(t+1) = \big[I_n \otimes A_i + (I_n \otimes B_{v,i})L(I_n \otimes C_i)\big]x(t) + B_u u(t) = Ax(t) + B_u u(t). \tag{2.4}$$

We impose the matrix A to be Schur stable, i.e.,

$$|\gamma| \leq 1, \quad \forall \gamma \in \sigma(A). \tag{2.5}$$

Remark 2 *Some observations are in order.*

- *The stability constrain can typically be satisfied by taking each of the interconnected subsystems sufficiently far from instability.*
- *The internal dynamics of each subsystem can conveniently be thought as a dynamics on subgraph itself, where the matrix A_i is the adjacency matrix of the subgraph and $B_{v,i}$, $B_{u,i}$ the matrices describing how the input from other subgraphs or from the external environment act on each of the nodes of the subsystem.*
- *We may alternatively want to order the overall state vector as*

$$x(t) = \big[x_{11}(t) \quad \dots \quad x_{n1}(t) \quad x_{12}(t) \dots x_{n2}(t) \quad \dots \quad x_{1n}(t) \quad \dots \quad x_{nn}(t)\big]$$

where $x_{ij}(t)$ is the j-th component of the state vector of subsystem i.
The dynamics can be written in state space form by exchanging the order of the two factors in each of the Kronecker product of equation (2.2)*, ($M \otimes N \to N \otimes M$) and by properly changing the matrix L (typically this matrix can be written as Kronecker product and it is sufficient to exchange the order of the factors):*

$$\Sigma : \begin{cases} x(t+1) = (A_i \otimes I_n)x(t) + (B_{v,i} \otimes I_n)v(t) + B_{u,i} \otimes \tilde{B}_u u(t) \\ y(t) = (C_i \otimes I_n)x(t) \end{cases} \tag{2.6}$$

The overall state equation can be written as:

$$x(t+1) = \big[A_i \otimes I_n + (B_{v,i} \otimes I_n)L(C_i \otimes I_n)\big]x(t) + B_{u,i} \otimes \tilde{B}_u u(t) = Ax(t) + B_u u(t) \tag{2.7}$$

With this alternative formulation we are just changing the basis for the overall state space. We will describe better in the next chapter with the help of an example the meaning of the two alternative notations.

We give now an example of how it is possible to describe a system of interconnected homogeneous components thanks to the introduced notation.

Example 5 (Interconnected systems)
Consider a system on network composed by 2 subsystems, each of them with state $x_i(t) \in \mathbb{R}^2$, the two are connected as shown in figure 2.1. The structure of a single subsystem is depicted in figure 2.2. Each subsystem is identical, we can see them as composed by two nodes (as described by remark 2), both nodes have a self-loop, moreover the first node of the subsystem has an input from ($v_i \in \mathbb{R}$) and an output to ($y_i \in \mathbb{R}$) other subsystems. The input from the external environment is $u_i(t) \in \mathbb{R}$ and it only enters the fist node of the subsystem.
The matrices describing each homogeneous subsystem with notation of equation (2.2) are hence:

$$A_i = \begin{bmatrix} 0.3 & 0 \\ 0.1 & 0.3 \end{bmatrix} \quad B_{v,i} = \begin{bmatrix} 0.2 \\ 0 \end{bmatrix} \quad C_i = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad B_{u,i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.8}$$

The state vector of the overall system is $x(t) = \begin{bmatrix} x_{11}(t) & x_{12}(t) & x_{21}(t) & x_{22}(t) \end{bmatrix} \in \mathbb{R}^4$, the input vector from other nodes is $v(t) = \begin{bmatrix} v_1(t) & v_2(t) \end{bmatrix} \in \mathbb{R}^2$ and the output vector is $y(t) = \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix} \in \mathbb{R}^2$.
We recall that the matrix L describes how the outputs from each subsystem are connected to inputs of other subsystems. Since only $y_1(t)$ is connected to $v_2(t)$, the matrix L can be written as:

$$L = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{2.9}$$

The matrix $\tilde{B}_u$ reads, by considering the input from the external environment only enters the first subsystems:

$$\tilde{B}_u = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.10}$$

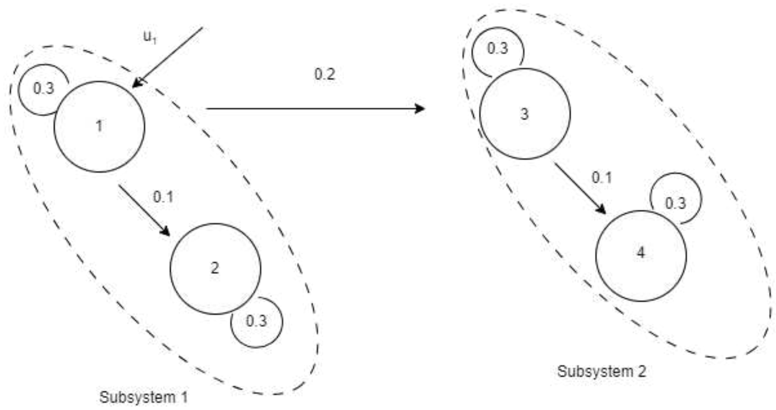

Figure 2.1: Example of interconnected system: overall system

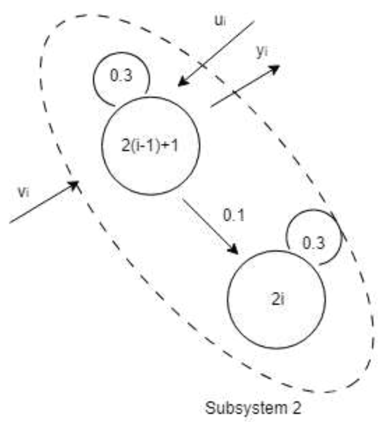

Figure 2.2: Example of interconnected system: single subsystem

Note that, with this definition of L, some inputs and outputs of the subsystems are not connected to any other subsystem (e.g., y_2 in not connected to v_1). The matrices A and B_u of the overall system can be written thanks to Kronecker products as defined in equation (2.4)*:*

$$A = \big[I_n \otimes A_i + (I_n \otimes B_{v,i})L(I_n \otimes C_i)\big] = \begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0.1 & 0.3 & 0 & 0 \\ 0.2 & 0 & 0.3 & 0 \\ 0 & 0 & 0.1 & 0.3 \end{bmatrix} \tag{2.11}$$

$$B_u = \tilde{B}_u \otimes B_{u,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{2.12}$$

2.2 Building homogeneous lifts

We describe now how it is possible to lift the aforementioned underlying network.
We consider now a graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ where $\hat{\mathcal{V}}$ is the set of nodes and $\hat{\mathcal{E}}$ the set of edges. This graph will represent the "expanded" or lifted graph associated to the "simple" or original graph $\mathcal{G}$.
By considering the same notation used in the definition of lifted graph in the preliminary section, and the non expanded network represented by $\mathcal{G}$ of the previous section, we will now describe how it is possible to perform a lift which is local and homogeneous (all the extend subsystems are governed by the same dynamical equations), hence obtained by lifting each node of the network in the same manner by expanding its internal state.
We design a map $\zeta : \hat{\mathcal{V}} \to \mathcal{V}$ which associate to each node in $\mathcal{V}$ k nodes in $\hat{\mathcal{V}} = \{1, 2 \dots, nk\}$.

Remark 3 *If the nodes in $\mathcal{V}$ have state vector of dimension N, each of the kn nodes in $\hat{\mathcal{V}}$ needs to have state of the same dimension.*
For sake of simplicity we reorder the nodes in the set $\hat{\mathcal{V}}$ so that $\zeta^{-1}(i) = \{(i-1)k+1, \dots ik\}$.
We can hence think to the network as composed by n subsystems of k nodes and the state of the network at time $t \in \mathbb{N}$ is described by the vector $x_e(t) = \begin{bmatrix} x_{e,1} & x_{e,2} & \dots & x_{e,n} \end{bmatrix} \in \mathbb{R}^{knN}$ where $x_{e,i}(t) \in \mathbb{R}^{kN}$ is the state of the i-th subsystem, it groups all the states of the nodes associated with node $i \in \mathcal{V}$.
We consider now the subsystem:

$$\Sigma_{e,i} : \begin{cases} x_{e,i}(t+1) = A_{e,i}x_{e,i}(t) + B_{e,v,i}v_i(t) + B_{e,u,i}u_i(t) \\ y_{e,i}(t) = C_{e,i}x_{e,i}(t) \end{cases} \tag{2.13}$$

Where

- $A_{e,i} \in \mathbb{R}^{kN \times kN}$ is the state matrix of the subsystem;
- $B_{e,v,i} \in \mathbb{R}^{kN \times q}$ is the input matrix from other nodes;
- $B_{e,u,i} \in \mathbb{R}^{kN \times 1}$ is the input matrix from the external environment;
- $C_{e,i} \in \mathbb{R}^{p \times N}$ is the output matrix of the subsystem.

By considering all the subsystems identical. i.e. $A_{e,i}$, $B_{e,v,i}$, $B_{e,u,i}$, $C_{e,i}$ are independent on i, the overall expanded system can be described by exploiting the usual Kronecker products as:

$$\Sigma_e : \begin{cases} x_e(t+1) = (I_n \otimes A_{e,i})x_e(t) + (I_n \otimes B_{e,v,i})v_e(t) + B_{e,u}u_e(t) \\ y_e(t) = (I_n \otimes C_i)x_e(t) \end{cases} \tag{2.14}$$

Where:

$$B_{e,u} = \tilde{B}_{e,u} \otimes B_{e,u,i} \in \mathbb{R}^{knN \times m} \qquad v_e(t) = L_e y_e(t) \tag{2.15}$$

The matrix $\tilde{B}_{e,u} \in \mathbb{R}^{n\times m}$ maps an input to a specific subsystem, hence its columns are canonical vectors.
The overall state equation reads:

$$\begin{aligned} x_e(t+1) &= \big[I_n \otimes A_{e,i} + (I_n \otimes B_{v,i})L(I_n \otimes C_{e,i})\big]x_e(t) + B_{e,u}u(t) = \\ &= A_e x_e(t) + B_{e,u}u_e(t) \end{aligned} \tag{2.16}$$

We take inspiration from Lifted Markov Chains, the state matrix A_e needs to respect the locality constraints on $\hat{\mathcal{G}}$ induced by the underlying graph $\mathcal{G}$. In other words if there was an edge between nodes $i, j \in \mathcal{V}$ on the original graph, then there must exist a connection between subsystems $\Sigma_{e,i}$ and subsystem $\Sigma_{e,j}$.
This constraint can be satisfied by imposing $y_{e,i} \in \mathbb{R}^p$ (as y_i), $v_{e,i} \in \mathbb{R}^q$ (as v_i) and by choosing $L = L_e$.
As for the simple network we impose the matrix A_e (and hence the overall system) to be Schur stable.

Remark 4 *The state matrix of the expanded subsystem describes both the internal dynamics of each of the k nodes composing the subsystem and how these nodes are connected each other. The matrix $B_{e,v,i}$ and $B_{e,u,i}$ instead describe how the inputs from other subsystems and from the external environment enter the nodes composing the subsystem. If the node in the simple system was connected to a certain input/output we are not obliged to connect each node composing the expanded subsystem to the corresponding input/output, however in order to respect the locality constraints on the lifted network there must exist at least one node connected to that input/output.*

Remark 5 *If the state of each node in the simple system has dimension $N > 1$ when performing lift we associate to this node a set of k nodes in the lifted graph each of them with state of dimension N (as described by the remark 3). Consequently, a decision needs to be made regarding the specific internal dynamics to be applied to these nodes. The most logical approach is to enforce them to adopt the identical internal dynamics as the original non-lifted node. By making this selection of the internal dynamics, there still remains some flexibility in determining how these nodes are interconnected with the other $k-1$ which are part of the same subsystem on the lifted network, with other subsystems and with the external environment (assuming the locality constraints are respected).*

Now we would like to "simulate" the marginalization operation that in Lifted Markov chains was used to get the distribution over the underlying graph from the distribution over the lifted graph, we are hence searching for a local linear operation which allow us to collapse the state of each expanded subsystem to the state of the corresponding simple system. The state of of the underlying network generated by the lifted dynamics will hence be:

$$x_I(t) = \tilde{C}x_e(t), \tag{2.17}$$

where $\tilde{C}$ is the matrix describing the collapsing map. In particular, in the following, we will always take the matrix $\tilde{C}$ so that $x_{I,i}(t) = \sum_{j=1}^{k} x_{e,ij}(t),\ \forall i$, where $x_{e,ij}(t)$ is the state of the j-th node associated to the i-th subsystem of the simple network, hence:

$$\tilde{C} = I_n \otimes \mathbb{1}_k^T \otimes I_N. \tag{2.18}$$

$x(t)$ may be visualized as the vector aggregation of an extra output from each subsystem whose relation with $x_e(t)$ is described by the entries of $\tilde{C}$.
It is also useful to define the local collapsing map expressing in matrix form the equation $x_{I,i}(t) = \sum_{j=1}^{k} x_{e,ij}(t)$:

$$x_{I,i}(t) = \tilde{C}_{e,i}x_{e,i}(t) = (\mathbb{1}_k^T \otimes I_N)x_{e,i}(t). \tag{2.19}$$

We will see in the next two chapters that a simple way to perform lift in order to add local memory to the evolution whenever each subsystem has the same number of inputs from other units and outputs is to introduce as many copies of node $i \in \mathcal{V}$ as the number of inputs from neighboring nodes. Each of these copies is then connected to a single input and a single output of the subsystem. Moreover we allow the possibility to create some interconnections between the copies of the same node (the matrix A_e does not need to be block diagonal).

Remark 6 *Similarly to remark 2 we may want to adopt a different notation and order the state vector of the overall system as:*

$$x_e(t) = \begin{bmatrix} x_{111}(t) & \dots & x_{n11}(t) & x_{121}(t)\dots & x_{n21}(t) & \dots & x_{1nk}(t) & \dots & x_{nnk}(t), \end{bmatrix}$$

where $x_{ijl}(t)$ is the j-th component of the state vector of the l-th node associated to the simple subsystem i.
In this case the overall system can be written by exchanging the order of Kronecker products and by properly changing matrix L and $\tilde{B}_u$ as:

$$\Sigma_e : \begin{cases} x_e(t+1) = (A_{e,i} \otimes I_n)x_e(t) + (B_{e,v,i} \otimes I_n)v_e(t) + B_{e,u,i} \otimes \tilde{B}_{e,u}u_e(t) \\ y_e(t) = (C_{e,i} \otimes I_n)x_e(t) \end{cases} \tag{2.20}$$

The overall state equation reads:

$$\begin{aligned} x_e(t+1) &= \left[A_{e,i} \otimes I_n + (B_{v,i} \otimes I_n)L(C_{e,i} \otimes I_n)\right]x_e(t) + B_{e,u,i} \otimes \tilde{B}_{e,u}u_e(t)u(t) = \\ &= A_e x_e(t) + B_{e,u}u_e(t). \end{aligned} \tag{2.21}$$

2.3 Performance comparison and constraints on the allowed dynamics

In this section we describe how it is possible to perform a comparison between the energy needed to control the state $x(t)$ of the underlying network through lifted dynamics or nominal dynamics. The approach used to control the simple network through the lift is to feed the expanded system with a control input u and let it evolve according to the dynamics described in state space form in equation (2.14). The state $x_I(t)$ induced by the lift is then given by equation (2.17).
As said before it is possible to consider $x_I(t)$ as an extra output of the expanded system, what we really want to control is this output. Therefore, the controllability definition which fits with the required task is output controllability. The metrics identified for energy related controllability analysis need to be applied not directly to the Gramian of the expanded system but to the output controllability Gramian. In particular once computed the controllability Gramian ($\mathcal{W}_e$) of the expanded system, it is necessary to consider the output controllability Gramian related to $x_I(t)$. Hence the correct Gramian to be used is:

$$\mathcal{W}_e^O = \tilde{C}\mathcal{W}_e\tilde{C}^T. \tag{2.22}$$

In particular, since we are interested in controlling **asymptotically** the system's state to the target one we need to consider infinite horizon controllability Gramians:

$$\mathcal{W} = \sum_{k=0}^{\infty} A^k B_u B_u^T (A^k)^T \quad \mathcal{W}_e^O = \tilde{C}\sum_{k=0}^{\infty} A_e^k B_{e,u} B_{e,u}^T (A_e^k)^T \tilde{C}^T. \tag{2.23}$$

We will then compare the two quantities $\lambda_{\min}(\mathcal{W})$ and $\lambda_{\min}(\mathcal{W}_e^O)$. The expanded dynamics is convenient in term of control energy if

$$\lambda_{\min}(\mathcal{W}) < \lambda_{\min}(\mathcal{W}_e^O) \tag{2.24}$$

We will consider also the two metrics $\text{trace}(\mathcal{W}^{-1})$ and $\text{trace}((\mathcal{W}_e^O)^{-1})$ in this case the expanded dynamics is convenient in term of control energy if

$$\text{trace}(\mathcal{W}^{-1}) > \text{trace}((\mathcal{W}_e^O)^{-1}) \tag{2.25}$$

As we are interested in assessing the advantages of the lifted network when the number of nodes grows, we will not directly compare the two metrics applied to the systems but their rate of growth. The latter are also called **asymptotic rates**. In the next chapters we will describe how to conduct the comparison both analytically and numerically (by performing simulations). Before proceeding with the computation of the metrics we need to identify some constraints which makes the analysis significant. We would like our lift to "act passively", as the advantages needs to be given by the structure of the lift and not by giving the system extra control power. We hence impose the following constraints:

- **number of control nodes and signals**: the expanded system needs to have the same number of input signals of the underlying graph.
- **locality of the control nodes** We need to respect the locality induced by the underlying graph, hence the i-th expanded subsystem is fed with an input iff the i-th simple system has an input (from the external environment).
- **non amplifying behaviour**: the controllability metrics we will consider are related with the energy of the control signals that enter the network. If the number of control nodes remains constant, as imposed by the first constraint, a network which is more difficult to control requires a control signal with higher control energy. If we allow subsystems in the lifted network to amplify signals more than what the related simple subsystem does for sure the expanded system would require signals with less energy.

We need now to understand how these constraints can be practically included in the model. The first two constraints can be embedded by imposing in equations (2.2) and (2.14):

$$\tilde{B}_{e,u} = \tilde{B}_u \tag{2.26}$$

As far as the last constraint is concerned, we would like that if the node and the expanded node are fed with the same input signals, either from the external environment or from other nodes, each of these signals is not amplified in terms of the ℓ_2-norm by the lifted node dynamics more than what the simple node dynamics does.
Recalling that each lifted subsystem and simple one have the same number of inputs, let $x_{I,i,j,k}(\text{t})$ be the k-th component of the induced state by the i-th subsystem when we consider only the j-th input acting on it, (all the other are considered to be 0) and let $x_{i,j,k}(\text{t})$ be the k-th component of the state of the simple system when only the j-th input act on it. What want to impose is:

$$||x_{I,i,j,k}(t)||_2^2 \leq ||x_{i,j,k}(t)||_2^2 \ \ \forall j, \forall i, \forall k \tag{2.27}$$

When the inputs are step signals this constraint can be achieved at least asymptotically by comparing the asymptotic gains of the two subsystems considering as output of the expanded system the induced state and of the simple system its state. From now on we will always assume step signals as input.
The open loop transfer function of interest for the simple system can be computed using the notation of equation (2.1) as:

$$W_i(z) = (zI_n - A_i)^{-1} \begin{bmatrix} B_{v,i} & B_{u,i} \end{bmatrix}. \tag{2.28}$$

The asymptotic gain is given by:

$$W_i(1) = \lim_{z \to 1} W_i(z) = (I_n - A_i)^{-1} \begin{bmatrix} B_{v,i} & B_{u,i} \end{bmatrix}. \tag{2.29}$$

The open loop transfer function of the expanded system considering as output the induced state can be computed with respect to the notation of equation (2.13) as:

$$W_{e,i}(z) = \tilde{C}_{e,i}(zI_n - A_{e,i})^{-1} \begin{bmatrix} B_{e,v,i} & B_{e,u,i} \end{bmatrix} = \tilde{C}_{e,i} \frac{\mathrm{adj}(zI_n - A_{e,i})}{\det(zI_n - A_{e,i})} \begin{bmatrix} B_{e,v,i} & B_{e,u,i} \end{bmatrix}. \tag{2.30}$$

The asymptotic gain is given by:

$$W_{e,i}(1) = \lim_{z \to 1} W_{e,i}(z) = \tilde{C}_{e,i} \frac{\mathrm{adj}(I_n - A_{e,i})}{\det(I_n - A_{e,i})} \begin{bmatrix} B_{e,v,i} & B_{e,u,i} \end{bmatrix}. \tag{2.31}$$

Now we can express the condition on the asymptotic gain in a compact form, we impose:

$$|W_{e,i}(1)| \leq |W_i(1)|, \tag{2.32}$$

where $|M|$ is the matrix whose entry $|M|_{ij}$ is the absolute values of M_{ij} and $M \leq N$ means that $M_{ij} \leq N_{ij}$, $\forall i, j$.
Another constraint that can be considered is on the stability of the expanded and simple systems. It is possible to require each expanded subsystem to be farther or at least as far as from instability than the corresponding simple node. To do so we impose the spectral radius of the matrix $A_{e,i}$ (matrix which describes the internal dynamics of the expanded subsystems) to be smaller than the spectral radius of A_i. Namely, let $\rho(M) = \max\{|\gamma|, \gamma \in \sigma(M)\}$, we impose:

$$\rho(A_{e,i}) \leq \rho(A_i). \tag{2.33}$$

If the overall network simple network is stable, by imposing the constraint on the spectral radius of expanded subsystems the lifted network should be stable (the relation with the stability of the overall network is highlighted in some simulations in the next two chapters). This constraint is also closely related with the amplifying behaviour of the system and in some cases implies that the condition in equation (2.27) is satisfied not only asymptotically but also in the transient. The relation is formally proved in the next chapters for the specific cases of line and grid networks. In particular, we will always impose both the constraints on asymptotic gain and spectral radius and we will prove that in all the cases we will consider for analytical computations and simulations, the constraint on the spectral radius is more restrictive than the constraint on the asymptotic gain, hence whenever we satisfy the former, the latter is satisfied.

Chapter 3

The Line Network Case

In this chapter we apply the methodologies developed in the preceding chapter to one of the most well-studied network topologies: the line network.

3.1 Model description for simple and lifted lines

3.1.1 Non-lifted network and control placement

In this section we describe the model of simple line network we will use in the rest of the thesis. We consider the directed line of n nodes depicted in figure 3.1. To the network it is associated a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{i = 1, ..., n\}$ is the set of nodes and $\mathcal{E}$ is the set of edges.
The state of the network is represented by the vector $x(t) \in \mathbb{R}^n$ whose entry $x_i(t) \in \mathbb{R}$ is the state of node $i \in \mathcal{V}$ at time $t \in \mathbb{N}$; for sake of simplicity, we are hence considering each node to have scalar state, with the notation of previous chapter the dimension of the internal state of each node of the network is $N = 1$.

We describe now in detail the topology of a general bidirectional line network:

- **Each node** $i \in \mathcal{V}$, except the first and the last one, has two outgoing edges one connecting i with $i + 1$ and the other one connecting i with $i - 1$, the weights assigned to these edges are respectively $b_{r,i+1}$ and $b_{l,i-1}$. The incoming edges from node $i + 1$ and from $i - 1$ are labeled as $b_{l,i}$ and $b_{r,i}$.

- The **first node** has only a single outgoing edge to node 2 and an incoming edge from node 2.

- The **last node** has only an outgoing edge to $n - 1$ and an incoming edge from $n - 1$.

- Each node $i \in \mathcal{V}$ has a **self loop** with weight λ_i.

- The weight associated to the input from the external environment to node i is labeled as $b_{u,i}$.

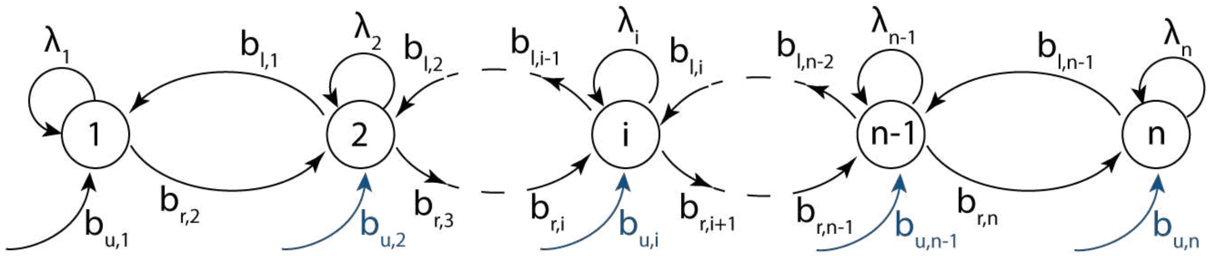

Figure 3.1: Simple line network

We now focus on the more convenient description (for our purposes) of the network as an interconnection of subsystems made of single nodes. The prototype single node system we will consider is depicted in figure 3.2. Each node is considered to have the same number of inputs and outputs as the number of incoming and outgoing edges described above. We label the input from node $i-1$ as $v_{r,i}(t)$ and the input from node $i+1$ as $v_{l,i(t)}$.
The weight associated to the edge $(j,i) \in \mathcal{E}$ is assigned to the corresponding input of node i (i.e., $v_{r,i}(t)$ and $v_{l,i}(t)$ affect the system with weights respectively $b_{r,i}$ and $b_{l,i}$), the outputs are considered to have unit weights, i.e. the outputs are exactly equal to the node state. Let $v_i(t) = \begin{bmatrix} v_{l,i(t)} & v_{r,i}(t) \end{bmatrix}^T$ be the vector whose entries are the inputs from other nodes as described above, let $u_i(t) \in \mathbb{R}$ be the input from the external environment, $y_i(t) = \begin{bmatrix} y_{l,i}(t) & y_{r,i}(t) \end{bmatrix}^T$ the vector of the outputs; the system can be written in state space form as:

$$\Sigma_i = \begin{cases} x_i(t+1) = A_i x_i(t) + B_{v,i} v_i(t) + b_{u,i} u_i(t) \\ y_i(t) = C_i x_i(t) \end{cases} \tag{3.1}$$

The matrices representing the system are:

$$A_i = \lambda_i \in \mathbb{R}, \quad B_{v,i} = \begin{bmatrix} b_{l,i} & b_{r,i} \end{bmatrix} \in \mathbb{R}^{1\times 2}, \quad b_{u,i} \in \mathbb{R}, \quad C_i = \mathbb{1}_2 \in \mathbb{R}^{2\times 1}. \tag{3.2}$$

In the upcoming section of this chapter, we will discuss the constraint on the lifted dynamics, to this aim it is convenient to compute the open loop transfer function of the single node system considering its state as output:

$$W_i(z) = (z - A_i)^{-1} B_i = \frac{B_i}{s - \lambda_i} = \begin{bmatrix} \frac{b_{l,i}}{z-\lambda_i} & \frac{b_{r,i}}{z-\lambda_i} & \frac{b_{u,i}}{z-\lambda_i}. \end{bmatrix} \tag{3.3}$$

The asymptotic gain is given by:

$$K_i = \lim_{z\to 1} W_i(z) = \begin{bmatrix} \frac{b_{l,i}}{1-\lambda_i} & \frac{b_{r,i}}{1-\lambda_i} & \frac{b_{u,i}}{1-\lambda_i}. \end{bmatrix} \tag{3.4}$$

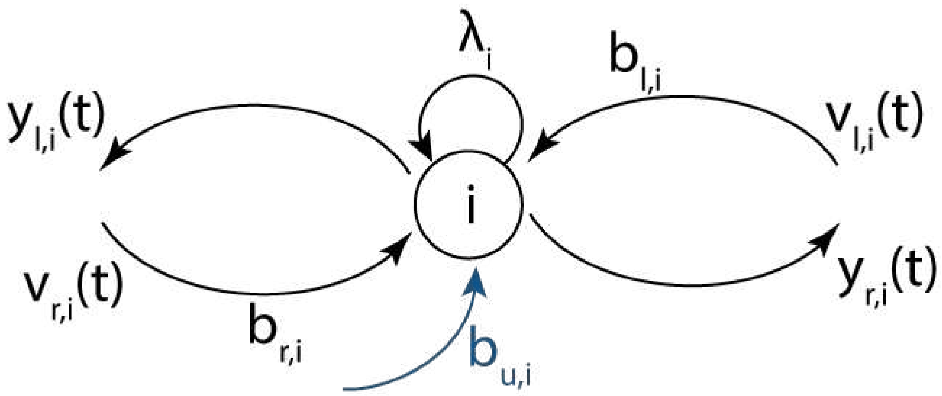

Figure 3.2: Simple line subsystem

By simplifying the model to assume all agents are identical (as outlined in the preceding chapter), we can leverage Kronecker products to formulate the state space equation for the entire interconnected system according to equation (2.6).
Recall that to make all the subsystems identical we need to impose in equation (3.2):

$$b_{l,i} = b_l, \quad b_{r,i} = b_r, \quad b_{u,i} = b_u, \quad \lambda_i = \lambda, \quad \forall i. \tag{3.5}$$

The overall system equation in state space form reads:

$$\Sigma : \begin{cases} x(t+1) = (A_i \otimes I_n)x(t) + (B_{v,i} \otimes I_n)v(t) + B_u u(t) \\ y(t) = (C_i \otimes I_n)x(t) \end{cases} \tag{3.6}$$

where:

$$B_u = b_u \tilde{B}_u, \qquad B_u \in \mathbb{R}^{n\times m}, \qquad v(t) = Ly(t). \tag{3.7}$$

In order to write the matrix L which describes the interconnections between different subsystems it is convenient to introduce the shift-to-left and shift-to-right matrices of dimension k, respectively $S_{k,l} \in \mathbb{R}^{k\times k}$ and $S_{k,r} \in \mathbb{R}^{k\times k}$, defined as:

$$[S_{k,l}]_{ij} = \delta_{i,j+1}, \qquad [S_{k,r}]_{ij} = \delta_{i+1,j}, \qquad \delta_{ij} = \begin{cases} 1 \ if \ i = j \\ 0 \ if \ i \neq j \end{cases} \tag{3.8}$$

Thanks to the introduced notions the matrix L reads:

$$L = e_{1,2}e_{1,2}^T \otimes S_{n,l} + e_{2,2}e_{2,2}^T \otimes S_{n,r} \tag{3.9}$$

Up to now, we've presented a general model of a controlled line network that enables us to provide a distinct input from the external environment to each node. However, in the subsequent sections of this chapter, our focus shifts towards examining how the process of lifting could enhance the controllability characteristics of the entire system, in particular when only few control inputs are available.
With this in mind we choose to feed the network with a single input $u(t) \in \mathbb{R}$ to node 1, the matrix $\tilde{B}_u$ should hence be selected in the following manner:

$$\tilde{B}_u = e_{1,n}, \tag{3.10}$$

where $e_{1,n}$ is the first canonical vector of dimension n.

Remark 7 *The chosen matrix $\tilde{B}_u$ implies the control inputs of nodes $j \in \{2, \ldots, n\}$ to be always disabled even thought they are present in the single system model. This modelling choice is necessary to guarantee the homogeneity of all the subsystems.*
The explicit expression of the vector $v(t)$ in function of L allows us to formulate the comprehensive state equation for the simple network as follows:

$$x(t+1) = \big[A_i \otimes I_n + (B_{v,i} \otimes I_n)L(C_i \otimes I_n)\big]x(t) + B_u u(t) = Ax(t) + B_u(t) \tag{3.11}$$

To conclude this subsection it is convenient to explicitly write the matrices A and B_u to highlight the structure of the system and the result of Kronecker product operations:

$$A = \begin{bmatrix} \lambda & b_l & 0 & \dots & 0 \\ b_r & \lambda & b_l & \dots & 0 \\ \vdots & b_r & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & b_l \\ 0 & 0 & 0 & b_r & \lambda \end{bmatrix}, \qquad B_u = \begin{bmatrix} b_u \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.12}$$

In the next sections we will often consider the symmetric configuration of the bidirectional line network which can be obtained by setting the parameters in the following way:

$$b_l = b_r = b. \tag{3.13}$$

With this choice of the parameters the state equation of the system can be written according to equation (3.11) as:

$$x(t+1) = Ax(t) + Bu(t) = \begin{bmatrix} \lambda & b & 0 & \dots & 0 \\ b & \lambda & b & \dots & 0 \\ \vdots & b & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & b \\ 0 & 0 & 0 & b & \lambda \end{bmatrix} x(t) + \begin{bmatrix} b_u \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(t). \tag{3.14}$$

3.1.2 The lifted line network

We will now delve into the process of performing the lift operation for the line network. As previously mentioned, lifting involves expanding the internal state of each node by introducing k additional vertices within the lifted graph for every node in the original, non-lifted graph. Consequently, to each node in the non-lifted graph it is associated a subsystem composed of k vertices in the lifted one. Through the lift operation, our objective is to enrich the internal dynamics of every node of the non-lifted network by introducing a localized memory of the direction from which the input is received. Our goal is to exploit the memory to transmit this input in the appropriate direction (possibly in the opposite direction to its point of origin), facilitating its transfer to distant nodes.

In the specific scenario of the line network, each node (excluding nodes 1 and n) can receive an input from either its left neighbor or its right neighbor. Consequently, the input has the potential to propagate in both the left and right directions. It is important for us to retain the information about the direction from which the control input is received.
To incorporate this information in our system, we can envision each subsystem in the lifted network as consisting of two nodes. The first node retains memory of the leftward direction, thus it receives input from the right neighbor and transmits it to the left neighbor. Conversely, the second node receives input from the left neighbor and propagates it to the right neighbor. Drawing inspiration from the Diaconis lift explained in the preliminary section, we also allow for the exchange of information between the two nodes within the same subsystem.

A different and more practical approach to understand the impact of this specific lift is to think the lifted network as composed by two distinct lines: the first line involves solely the nodes within each subsystem that transmit inputs to the left, thus constituting a left-oriented network; the second line includes exclusively the nodes within each subsystem that propagate inputs to the right.

In figure 3.3 it is depicted the lifted network we would like to build. The nodes on the lower side of the image constitute the right-oriented line, whereas the nodes on the upper part of the image are in the left-oriented line. The dotted lines highlight each subsystem composed by two nodes associated to a single node of the original network.

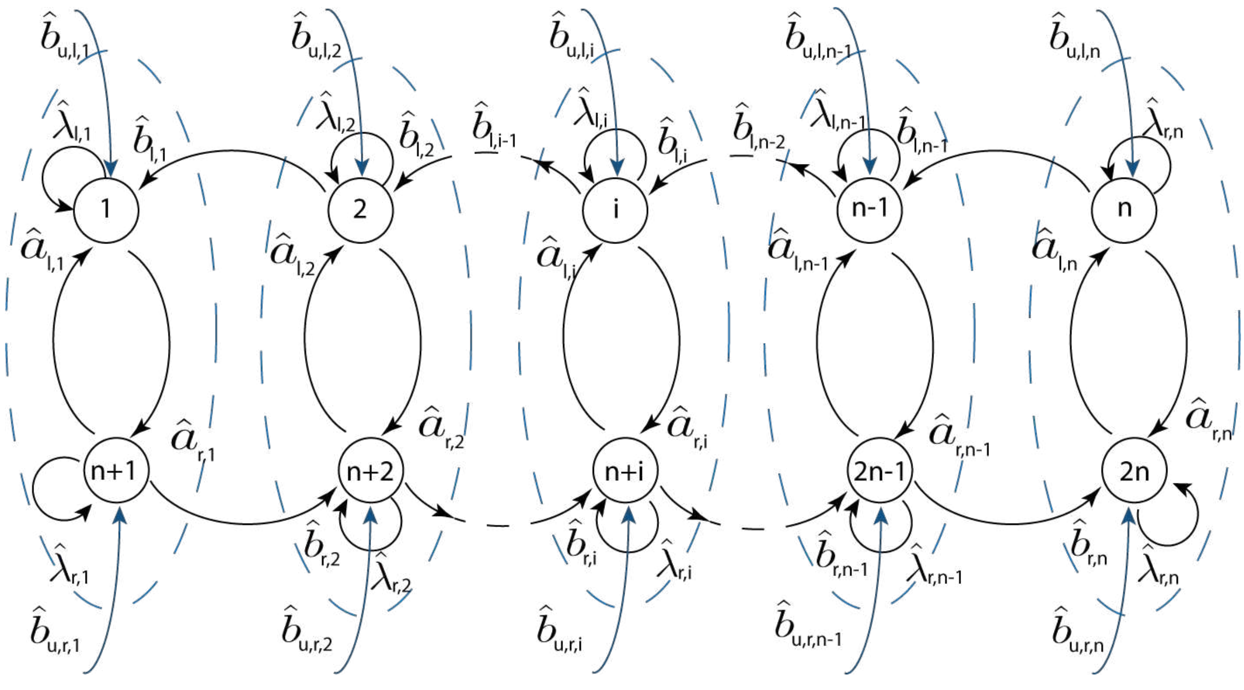

Figure 3.3: Lifted line network

Once explained the idea we have in mind we proceed to describe the lifted network according to the model formalized in the previous chapter.

We consider the lifted graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ describing the network in figure 3.3 where $\hat{\mathcal{V}}$ is the set of nodes and $\hat{\mathcal{E}}$ the set of edges. We design a map $\zeta : \hat{\mathcal{V}} \mapsto \mathcal{V}$ which associates to each node in $\mathcal{V}$ 2 nodes in $\hat{\mathcal{V}} = \{1, 2 \ldots, 2n\}$. We reorder the nodes in the set $\hat{\mathcal{V}}$ so that $\zeta^{-1}(i) = \{i, n+i\}$.

Remark 8 *We have defined a different map ζ^{-1} with respect to the one presented in the general procedure for the lift since we want to exploit the alternative formulation of the overall expanded system.*

We can hence think to the lifted network as composed by n subsystems of 2 nodes, the state of the network at time $t \in \mathbb{N}$ is described by the vector $x_e(t) = \begin{bmatrix} x_{e,1} & x_{e,2} & \ldots & x_{e,n} \end{bmatrix} \in \mathbb{R}^{2n}$ where $x_{e,i}(t) = \begin{bmatrix} x_{e,i,l}(t) & x_{e,i,r}(t) \end{bmatrix}$, $x_{e,i}(t) \in \mathbb{R}^2$, is the state of the i-th subsystem, $i \in \{1, ..., n\}$.
The structure of each subsystem is depicted in figure 3.4.
We label the external input as $u_{e,i}(t) \in \mathbb{R}$ and we consider the same input to enter nodes i and $n+i$ with weights respectively $\hat{b}_{u,i,l}$ and $\hat{b}_{u,i,r}$.
Let $v_{e,i}(t) = \begin{bmatrix} v_{e,i,l}(t) & v_{e,i,r}(t) \end{bmatrix}^T$ be the vector whose entries are the inputs from neighboring subsystems, $\hat{b}_{i,l}$ and $\hat{b}_{e,i,r}$ are the weights assigned respectively to inputs $v_{e,i,l}(t)$ and $v_{e,i,r}(t)$.
Let $y_{e,i}(t) = \begin{bmatrix} y_{e,i,l}(t) & y_{e,i,r}(t) \end{bmatrix}^T$ be the vector of the outputs from the subsystem, $y_{e,i,l}(t)$ is the the output from the first node, $y_{e,i,r}(t)$ is the output from the second node. The outputs are exactly equal to the corresponding node state, i.e., $y_{e,i,r}(t) = x_{e,i,r}(t)$ and $y_{e,i,l}(t) = x_{e,i,l}(t)$, $\forall t$.
We allow each node to have a self-loop, respectively the first and second nodes have self-loops $\hat{\lambda}_{l,i}$ and $\hat{\lambda}_{r,i}$. Furthermore, the first node is connected to the second with weight $a_{r,i}$ and the second to the first with weight $a_{l,i}$.

Using the formulation of equation (2.13) the coupled subsystem is described by the system of equation:

$$\Sigma_{e,i} : \begin{cases} x_{e,i}(t+1) = A_{e,i}x_{e,i}(t) + B_{e,v,i}v_i(t) + B_{e,u,i}u_i(t) \\ y_{e,i}(t) = C_{e,i}x_{e,i}(t) \end{cases} \tag{3.15}$$

where:

$$A_{e,i} = \begin{bmatrix} \hat{\lambda}_{l,i} & \hat{a}_{l,i} \\ \hat{a}_{r,i} & \hat{\lambda}_{r,i} \end{bmatrix}, \qquad B_{e,v,i} = \begin{bmatrix} \hat{b}_{l,i} & 0 \\ 0 & \hat{b}_{r,i} \end{bmatrix}, \qquad B_{e,u,i} = \begin{bmatrix} \hat{b}_{u,l,i} \\ \hat{b}_{u,r,i} \end{bmatrix}. \qquad C_{e,i} = I_2 \tag{3.16}$$

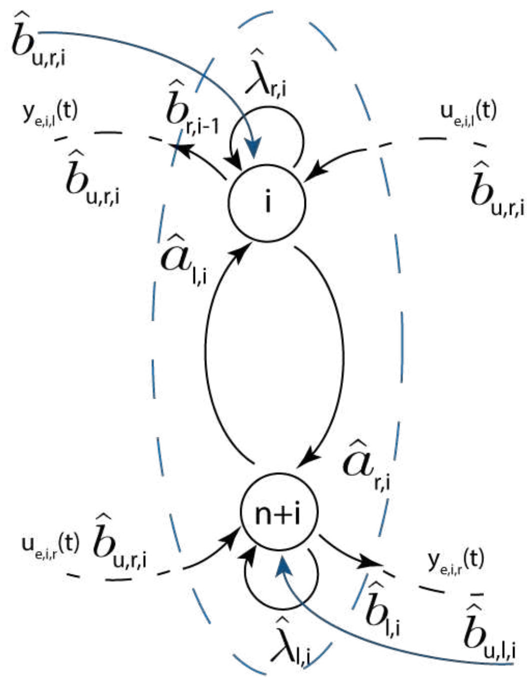

Figure 3.4: Lifted line subsystem

Observation: For the following computations it is useful to notice that the eigenvalues of A_e are:

$$\begin{aligned}\gamma_1 &= \frac{\lambda_{l,i}}{2} + \frac{\lambda_{r,i}}{2} + \frac{((\lambda_{l,i}-\lambda_{r,i})^2 + 4a_{l,i}a_{r,i})^{1/2}}{2}\\ \gamma_2 &= \frac{\lambda_{l,i}}{2} + \frac{\lambda_{r,i}}{2} - \frac{((\lambda_{l,i}-\lambda_{r,i})^2 + 4a_{l,i}a_{r,i})^{1/2}}{2}\end{aligned} \tag{3.17}$$

We now consider the whole system as the interconnection of single coupled systems.

We define the output vector for the overall system as

$$y_e(t) = \begin{bmatrix} y_{e,1,l}(t) & y_{e,2,l}(t) & \dots & y_{e,n,l}(t) & y_{e,1,r}(t) & \dots & y_{e,n,r}(t)\end{bmatrix}^T.$$

By considering all the subsystems $\Sigma_{e,i}$ identical and by using the alternative notation in equation (2.20), it is possible to write the overall system state equation. To make all the subsystems identical we need to impose the matrices in equation (3.16) to be independent on i, hence we set the parameters:

$$\hat\lambda_{l,i} = \hat\lambda_l, \quad \hat\lambda_{r,i} = \hat\lambda_r, \quad \hat a_{l,i} = \hat a_l, \quad \hat b_{l,i} = \hat b_l, \quad \hat b_{r,i} = \hat b_r, \quad \hat b_{u,r,i} = \hat b_{u,r}, \quad \hat b_{u,l,i} = \hat b_{u,l}, \quad \forall i. \tag{3.18}$$

The overall state equation for the expanded system can be written as:

$$\Sigma_e : \begin{cases} x_e(t+1) = (A_{e,i}\otimes I_n)x_e(t) + (B_{e,v,i}\otimes I_n)v_e(t) + B_{e,u,i}\otimes \tilde B_{e,u}u_e(t)\\ y_e(t) = (C_{e,i}\otimes I_n)x_e(t)\end{cases} \tag{3.19}$$

In order to satisfy the locality constraints typical of lift operation we impose:

$$\tilde B_{e,u} = \tilde B_u, \qquad L_e = L. \tag{3.20}$$

Coherently with the choices made for the simple system we are hence applying the external control input $u(t)\in\mathbb{R}$ only to the first subsystem. The overall state equation can be written according to equation (2.21) as:

$$\begin{aligned} x_e(t+1) &= \left[A_{e,i}\otimes I_n + (B_{v,i}\otimes I_n)L(C_{e,i}\otimes I_n)\right]x_e(t) + B_{e,u,i}\otimes\tilde B_{e,u}u_e(t) =\\ &= A_e x_e(t) + B_{e,u}u_e(t).\end{aligned} \tag{3.21}$$

It is useful to write explicitly the matrices A_e and B_e:

$$A_e = \begin{bmatrix} A_{e,ll} & A_{e,lr}\\ A_{e,rl} & A_{e,rr}\end{bmatrix} = \begin{bmatrix} \hat\lambda_l & \hat b_l & 0 & \dots & \hat a_l & 0 & \dots & 0\\ 0 & \hat\lambda_l & \ddots & \vdots & 0 & \hat a_l & \ddots & \vdots\\ \vdots & \ddots & \ddots & \hat b_l & \vdots & \ddots & \ddots & 0\\ 0 & \dots & 0 & \hat\lambda_l & 0 & \dots & 0 & \hat a_l\\ \hat a_r & 0 & \dots & 0 & \hat\lambda_r & 0 & \dots & 0\\ 0 & \hat a_r & \ddots & \vdots & \hat b_r & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0\\ 0 & \dots & 0 & \hat a_r & 0 & \dots & \hat b_r & \hat\lambda_r \end{bmatrix}, \qquad B_e = \begin{bmatrix}\hat b_{u,l}\\ \mathbf{0}\\ \hat b_{u,r}\\ \mathbf{0}\end{bmatrix}, \tag{3.22}$$

where $\mathbf{0}$ is a column vector of all zeros of dimension $n-1$.
We better clarify the structure of the matrix A_e:

- the matrix $A_{e,ll}\in\mathbb{R}^{n\times n}$ has $\hat\lambda_l$ on the diagonal and $\hat b_l$ on the upper diagonal;
- $A_{e,lr}\in\mathbb{R}^{n\times n}$ has $\hat a_l$ on the diagonal;
- $A_{e,rl}\in\mathbb{R}^{n\times n}$ has $\hat a_r$ on the diagonal;

- $A_{e,rr} \in \mathbb{R}^{n\times n}$ has $\hat{\lambda}_r$ on the diagonal and $\hat{b}_r$ on the lower diagonal.

A special configuration which will be considered in the following sections for numerical simulations and proofs is the symmetric one. It can be obtained by setting the variables describing the internal dynamics of each lifted subsystem in the following way:

$$\hat{b}_l = \hat{b}_r = \hat{b}, \qquad \hat{a}_l = \hat{a}_r = \hat{a}, \qquad \hat{\lambda}_l = \hat{\lambda}_r = \lambda, \qquad \hat{b}_{u,l} = \hat{b}_{u,r} = \hat{b}_u \tag{3.23}$$

The overall state equation for the symmetric lifted line network can be written according to (2.16) as:

$$x_e(t+1) = A_e x_e(t) + B_{e,u} u_e(t) = \begin{bmatrix} A_{e,ll} & A_{e,lr} \\ A_{e,rl} & A_{e,rr} \end{bmatrix} x_e(t) + \begin{bmatrix} \hat{b}_u \\ \mathbf{0} \\ \hat{b}_u \\ \mathbf{0} \end{bmatrix} u_e(t), \tag{3.24}$$

where:

$$A_{e,ll} = \begin{bmatrix} \hat{\lambda} & \hat{b} & & & 0 \\ & \hat{\lambda} & \hat{b} & & \\ & & \ddots & \ddots & \\ & & & \hat{\lambda} & \hat{b} \\ 0 & & & & \hat{\lambda} \end{bmatrix}, \quad A_{e,lr} = A_{e,rl} = \begin{bmatrix} \hat{a} & & & 0 \\ & \hat{a} & & \\ & & \ddots & \\ & & & \hat{a} & \\ 0 & & & & \hat{a} \end{bmatrix}, \quad A_{e,rr} = \begin{bmatrix} \hat{\lambda} & & & & 0 \\ \hat{b} & \hat{\lambda} & & & \\ & \ddots & \ddots & & \\ & & \hat{b} & \hat{\lambda} & \\ 0 & & & \hat{b} & \hat{\lambda} \end{bmatrix}. \tag{3.25}$$

Remark 9 *We have now a practical example to explain the differences between the two alternative descriptions of the system obtained by exchanging factors in Kronecker products. We compare the two equations for the expanded system* (2.14) *and* (2.20). *Using the first description by looking to the matrix A_e of the overall system the matrices on the diagonal of dimension k reveal the internal dynamics between the nodes of the same expanded subsystem whereas the other entries express the interconnection between different subsystems. The focus is hence on the description of the interaction between different subsystems composed of two nodes. Using the second notation instead the matrices on the diagonal of A_e of dimension n describe the internal dynamics between the nodes within the same copy of the line network (the left oriented one and right oriented one), whereas the off diagonal matrices the interconnections between the two lines. The focus this time is hence on the representation of the interactions between the two lines.*

It remains to describe how to perform the marginalization operation to collapse the state of the lifted network to that one of the simple network.
The state of node i of the underlying graph $\mathcal{G}$ induced by the lifted dynamics will be:

$$x_I(t) = \tilde{C} x_e(t). \tag{3.26}$$

We take the matrix $\tilde{C}$ so that $x_{I,i}(t) = \sum_{j=1}^{2} x_{e,ij}(t)$, hence:

$$\tilde{C} = \mathbb{1}_k^T \otimes I_N \otimes I_n = \begin{bmatrix} I_n & I_n \end{bmatrix}. \tag{3.27}$$

We define also the local collapsing map:

$$\tilde{C}_{e,i} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{3.28}$$

As done before for the simple node system, due to the constraints discussion we will perform in the next section, it is convenient to compute the transfer function from the inputs of the single expanded subsystem to the corresponding collapsed state:

$$\begin{aligned} W_{e,i}(z) &= \tilde{C}_{e,i}(zI_n - A_{e,i})^{-1} \begin{bmatrix} B_{e,v,i} & B_{e,u,i} \end{bmatrix} = \tilde{C}_{e,i} \frac{\operatorname{adj}(zI_n - A_{e,i})}{\det(zI_n - A_{e,i})} \begin{bmatrix} B_{e,v,i} & B_{e,u,i} \end{bmatrix} = \\ &= \tilde{C}_{e,i} \frac{\begin{bmatrix} \hat{b}_{l,i}(z-\hat{\lambda}_{r,i}) & \hat{a}_{l,i}\hat{b}_{r,i} & (z-\hat{\lambda}_{r,i})\hat{b}_{u,l,i} + \hat{a}_{l,i}\hat{b}_{u,r,i} \\ \hat{a}_{r,i}\hat{b}_{l,i} & (z-\hat{\lambda}_{l,i})\hat{b}_{r,i} & \hat{a}_{r,i}\hat{b}_{u,l,i} + (z-\hat{\lambda}_{l,i})\hat{b}_{u,r,i} \end{bmatrix}}{\hat{\lambda}_{l,i}\hat{\lambda}_{r,i} - \hat{a}_{l,i}\hat{a}_{r,i} - (\hat{\lambda}_{l,i} + \hat{\lambda}_{r,i})z + z^2} = \end{aligned} \tag{3.29}$$

$$= \frac{\left[\hat{b}_{l,i}(z-\hat{\lambda}_{r,i})+\hat{a}_{r,i}\hat{b}_{l,i} \quad \hat{a}_{l,i}\hat{b}_{r,i}+(z-\hat{\lambda}_{l,i})\hat{b}_{r,i} \quad (z-\hat{\lambda}_{r,i})\hat{b}_{u,l,i}+\hat{a}_{l,i}\hat{b}_{u,r,i}+\hat{a}_{r,i}\hat{b}_{u,l,i}+(z-\hat{\lambda}_{l,i})\hat{b}_{u,r,i}\right]}{\hat{\lambda}_{l,i}\hat{\lambda}_{r,i}-\hat{a}_{l,i}\hat{a}_{r,i}-(\hat{\lambda}_{l,i}+\hat{\lambda}_{r,i})z+z^2} \tag{3.30}$$

The asymptotic gain is given by:

$$K = \lim_{z\to 1} W_{e,i}(z) = \tag{3.31}$$

$$= \frac{\left[\hat{b}_{l,i}(1-\hat{\lambda}_{r,i})+\hat{a}_{r,i}\hat{b}_{l,i} \quad \hat{a}_{l,i}\hat{b}_{r,i}+(1-\hat{\lambda}_{l,i})\hat{b}_{r,i} \quad (1-\hat{\lambda}_{r,i})\hat{b}_{u,l,i}+\hat{a}_{l,i}\hat{b}_{u,r,i}+\hat{a}_{r,i}\hat{b}_{u,l,i}+(1-\hat{\lambda}_{l,i})\hat{b}_{u,r,i}\right]}{\hat{\lambda}_{l,i}\hat{\lambda}_{r,i}-\hat{a}_{l,i}\hat{a}_{r,i}-(\hat{\lambda}_{l,i}+\hat{\lambda}_{r,i})+1} \tag{3.32}$$

3.2 Discussion of the constraints

In this section we describe in detail how the constraints for the allowed dynamics identified in the previous chapter affects the proposed lift for the line network.
The first constraint we need to satisfy is on the **number of control nodes and signals**. The lifted network needs to have the same number of control inputs of the underlying non-expanded counterpart. Indeed the proposed lift has been thought already by considering this constraint, in fact since the non-expanded network has only one input signal we have considered the lifted network to have just one input.
The second constraint regards the **locality of the control nodes** and we have identified that a necessary condition for this constraint to be satisfied is to impose the matrix $\tilde{B}_u$ to be equal to the matrix $\tilde{B}_{e,u}$, this constraint has already been applied in the formulation of the lifted line. In particular, since the control input on the non-expanded system enters only the first node, we have imposed the control input of the lifted network to enter the first subsystem (which becomes the input subsystem). Since no constraints are placed on how the input signal enters the node of the control subsystem we have enabled the input signal to enter each of the two nodes with a possibly different amplification constant.
Finally an additional constraint which was already outlined in the preliminary section was to respect the locality of the non-expanded graph and this has been granted by choosing the matrix L_e of the overall expanded system to be equal to L.
Up to now we have not considered any constraints on the internal dynamics of the lifted subsystem hence we can choose each of the variables defined in equation (3.18) at our discretion. Of course we still need to consider the last constraint defined in the previous chapter, i.e., the **non amplifying behaviour** of lifted subsystems.
Each entry of the asymptotic gain matrix $W_{e,i}$ for the expanded subsystem needs to have modulus smaller than the corresponding entry of the asymptotic gain matrix W_i for the simple subsystem i.e.:

$$|W_{e,i}(1)| \leq |W_i(1)|. \tag{3.33}$$

We will consider also the constraint on the stability of subsystems, each eigenvalue of the matrix $A_{e,i}$ needs to have modulus smaller than the maximum eigenvalues of the matrix A_i which can be translated in terms of the spectral radius as:

$$\rho(A_{e,i}) \leq \rho(A_i). \tag{3.34}$$

To discuss these constraints we simplify the notation and reduce both the simple and lifted network to the symmetric case. In the rest of this chapter we will hence always consider symmetric networks. This choice is not only to reduce the degrees of freedom in the proposed model and ease computations but it is due to the idea that in these networks the control input spreads similarly in all the directions. By operating the lift we modify the internal dynamics of the nodes such that the input signal propagates faster in one direction to reach the farther subsystems. In the symmetric case hence we expect the lifted network to perform better in terms of control energy than the simple one.
With the symmetric assumption on the parameter choice the transfer function for the lifted system from the inputs to the collapsed state reads:

$$W_{e,i}(z) = \left[\frac{\hat{b}}{z-\hat{\lambda}-\hat{a}} \quad \frac{\hat{b}}{z-\hat{\lambda}-\hat{a}} \quad \frac{2\hat{b}_u}{z-\hat{\lambda}-\hat{a}}\right], \tag{3.35}$$

and the asymptotic gain is:

$$W_{e,i}(1) = \Big[\frac{\hat{b}}{1-\hat{\lambda}-\hat{a}} \quad \frac{\hat{b}}{1-\hat{\lambda}-\hat{a}} \quad \frac{2\hat{b}_u}{1-\hat{\lambda}-\hat{a}}\Big]. \tag{3.36}$$

Moreover the eigenvalues of $A_{e,i}$ are $\hat{\lambda}+\hat{a}$ and $\hat{\lambda}-\hat{a}$.
The transfer function for the simple system from the inputs to the state reads instead:

$$W_i = \left[\tfrac{b}{z-\lambda} \quad \tfrac{b}{z-\lambda} \quad \tfrac{b_u}{z-\lambda}\right], \tag{3.37}$$

and the asymptotic gain is:

$$W_i(1) = \Big[\frac{b}{1-\lambda} \quad \frac{b}{1-\lambda} \quad \frac{b_u}{1-\lambda}\Big]. \tag{3.38}$$

The following proposition establishes a relation between the constraints on the modulus of eigenvalues and on the asymptotic gain and gives us a method to choose the parameters such that both the constraints are satisfied.

Proposition 2 *Consider the symmetric line network with positive self loops described in equation* (3.14) *and the symmetric lifted line network of equation* (3.24)*, where we set* $\hat{b} = b$*. Let* $\hat{\gamma}_j$*,* $j \in \{1, 2\}$ *be the eigenvalues of* $A_{e,i}$ *and* λ *be the positive self loop of the simple network (i.e. the unique eigenvalue of* A_i*) where we impose* $\lambda < 1$*,* $|\gamma_j| < 1$*,* $\forall j$*. Let* $\hat{b}_u = \frac{b_u}{2}$ *then:*

$$|\gamma_j| \le |\lambda| \ \forall j \Longrightarrow |W_{e,i}(1)| \le |W_i(1)|. \tag{3.39}$$

Proof: We consider the two expressions of the asymptotic gains for the symmetric networks in equation (3.36) and (3.38). Let $\gamma_1 = \hat{\lambda}+\hat{a}$ then

$$\begin{aligned} |W_{e,i}(1)| \le |W_i(1)| &\Longleftrightarrow \left|\frac{b}{1-\hat{\gamma}_1}\right| \le \left|\frac{b}{1-\lambda}\right| \vee \left|\frac{2\hat{b}_u}{1-\hat{\gamma}_1}\right| \le \left|\frac{b_u}{1-\lambda}\right| \\ &\Longleftrightarrow \frac{1}{|1-\hat{\gamma}_1|} \le \frac{1}{|1-\lambda|} \\ &\Longleftrightarrow |1-\gamma_1| \ge |1-\lambda|. \end{aligned} \tag{3.40}$$

Since $|\gamma_1| < 1$ and $|\lambda| < 1$ then:

$$\frac{1}{|1-\hat{\gamma}_1|} \le \frac{1}{|1-\lambda|} \Longleftrightarrow 1-\gamma_1 \ge 1-\lambda \Longleftrightarrow \gamma_1 \le \lambda. \tag{3.41}$$

Hence if we impose $|\gamma_j| \le |\lambda|$, $\forall j$ then for sure $\gamma_1 \le \lambda$ and hence $|W_{e,i}(1)| \le |W_i(1)|$. ■

If we choose the weights according to the previous proposition then for sure the constraint on the asymptotic gain is verified, moreover since the transfer function in equations (3.35) and (3.37) are of first order systems if we choose positive self-loops in the simple network the non-amplifying behaviour holds also in the transient (we are consider step signals as input). However, we are requiring something more, i.e., the expanded system to be farther away from instability than the simple system. This is a logical requirement to guarantee the stability of the expanded network. If we impose the simple network to be stable and we apply the constraints on the eigenvalues to each subsystem of the lifted network we expect the overall expanded network to be stable. In the section about simulations we verify experimentally that the claim is correct.
In the next sections we will always choose the parameters according to the above proposition, not only to satisfy the constraints but because by choosing $\hat{b} = b$ we do not change how different nodes/subsystems interact with each other, moreover by setting $\hat{b}_u = \frac{b_u}{2}$ we equally split the input between the two nodes composing the expanded subsystem.

3.3 Analytic derivation of metric values

In the preceding section we have identified the set of constraints we need to impose to the lifted network to make our comparison meaningful. We are now ready to compute analytically the metrics in order to compare the performance achieved exploiting the lifted dynamics. For the analytic computation of the metrics we will again constrain to the specific case of symmetric networks. In this section we will only consider the analytic computation of the metric trace of the inverse of the Gramian for the two systems as the metric minimum eigenvalue can be proved to be closely related to it as the dimension of the network grows. We recall that the metric of interest requires the knowledge of the controllability Gramian. We need hence to compute the controllability Gramian $\mathcal{W}$ for the simple network and the output controllability Gramian $\mathcal{W}_e^O$ for the expanded network. With reference to the notation of previous sections the two Gramians are described by the following equations

$$\mathcal{W} = \sum_{k=0}^{\infty} A^k BB^T (A^k)^T, \qquad \mathcal{W}_e^O = \sum_{k=0}^{\infty} A_e^k B_e B_e^T (A_e^k)^T. \tag{3.42}$$

If the matrix A and A_e are stable to find the solution of these infinite sums we can exploit discrete-time Lyapunov equations. For the Gramian computation we solve:

$$A\mathcal{W}A^T - \mathcal{W} = -BB^T. \tag{3.43}$$

For the output controllability Gramian we first compute:

$$A_e \mathcal{W}_e A_e^T - \mathcal{W}_e = -B_e B_e^T. \tag{3.44}$$

The solution is then given by:

$$\mathcal{W}_e^O = \tilde{C}\mathcal{W}_e \tilde{C}^T. \tag{3.45}$$

However, this computational approach requires the solution of a system of equation of large dimension hence it can not be used to find explicit analytic expressions of the quantities of interest. Moreover to compute the metrics:

$$\text{trace}[\mathcal{W}^{-1}], \qquad \text{trace}[(W_e^O)^{-1}] \tag{3.46}$$

we need to perform an inverse of the Gramian matrices which is computationally expensive and typically ill conditioned. We aim to find an alternative explicit solution for the Gramians and the related matrices which does not require Lyapunov equations. The following proposition provides an expression to compute the inverse of the Gramian for the simple network.

Proposition 3 *Consider the symmetric line network described in equation* (3.14) *with single input on the first node and positive self-loops. Let matrix A of the system be stable. Consider the quantities:*

$$\gamma_h = \lambda + 2|b| \cos\left(\frac{h\pi}{n+1}\right), \quad d_{ii} = \frac{b_u \sin\left(\frac{i\pi}{n+1}\right)}{\sqrt{\frac{n+1}{2}}}. \tag{3.47}$$

Then

$$\text{trace}[\mathcal{W}^{-1}] = \sum_{i=1}^{n} \frac{(1-\gamma_i^2)}{d_{ii}^2} \prod_{k \neq i} \Big| \frac{1-\gamma_i\gamma_k}{\gamma_k - \gamma_i} \Big|^2. \tag{3.48}$$

Proof: The adjacency matrix of a bidirectional line network is a tridiagonal Toeplitz matrix. We consider as a reference [6]. A generic tridiagonal Toeplitz matrix of dimension n reads:

$$T = \begin{bmatrix} \delta & \tau & & & 0 \\ \sigma & \delta & \tau & & \\ & \sigma & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \tau \\ 0 & & & \sigma & \delta \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{3.49}$$

The matrix T has n simple eigenvalues if $\sigma\tau \neq 0$ which can explicitly be computed as:

$$\lambda_h(T) = \delta + 2\sqrt{\sigma\tau}\cos\frac{h\pi}{n+1}, \quad h = 1, \dots, n. \tag{3.50}$$

The adjacency matrix of the network we are considering is a tridiagonal Toeplitz matrix of dimension n with $\delta = \lambda, \tau = b, \sigma = b$, namely,

$$A = \begin{bmatrix} \lambda & b & 0 & \dots & 0 \\ b & \lambda & b & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & b \\ 0 & 0 & 0 & \dots & \lambda \end{bmatrix}. \tag{3.51}$$

A has n distinct eigenvalues since $b^2 \neq 0$, we label them as γ_h, $h = 1, \dots, n$. They can be explicitly computed as:

$$\gamma_h = \lambda + 2|b|\cos\left(\frac{h\pi}{n+1}\right). \tag{3.52}$$

The components of the right eigenvector $x_h = \begin{bmatrix} x_{h,1}, & x_{h,2} & \dots & x_{h,n} \end{bmatrix}^T$ associated with γ_h are:

$$x_{h,k} = \sin\left(\frac{hk\pi}{n+1}\right), \quad k = 1, \dots, n, \quad h = 1, \dots, n, \tag{3.53}$$

and the corresponding left eigenvector is $y_h = \begin{bmatrix} y_{h,1}, & \dots & y_{h,n} \end{bmatrix} = x_h^T$ since the matrix A is symmetric. A has n distinct eigenvalues hence it is diagonalizable, moreover if we choose b, λ such that A is Schur stable we can compute the infinite-horizon controllability Gramian.

The spectral radius of A is given by:

$$\rho(A) = \max\Big\{\Big|\lambda + 2|b|\cos\left(\frac{\pi}{n+1}\right)\Big|, \Big|\lambda + 2|b|\cos\left(\frac{n\pi}{n+1}\right)\Big|\Big\}. \tag{3.54}$$

In particular:

$$\rho(A) = \begin{cases} \lambda + 2|b|\cos\left(\frac{\pi}{n+1}\right) & \lambda \geq 0 \\ -\lambda - 2|b|\cos\left(\frac{\pi}{n+1}\right) & \lambda < 0 \end{cases} \tag{3.55}$$

A is stable iff $\rho(A) < 1$. if we impose positive self loops and fix λ, A is stable iff

$$|b| < \frac{(1-\lambda)}{2\cos(\pi/(n+1))}. \tag{3.56}$$

Let $\Lambda = \text{diag}(\gamma_1, \dots, \gamma_n)$ and V be a change of basis matrix such that $A = V\Lambda V^{-1}$, then the Gramian of the system can be written as:

$$\begin{aligned} \mathcal{W} &= \sum_{k=0}^{\infty} A^k BB^T (A^k)^T = \sum_{k=0}^{\infty} V\Lambda^k V^{-1} BB^T (V^{-1})^T (\Lambda^T)^k V^T = \\ &= V\Big[\sum_{k=0}^{\infty} V\Lambda^k V^{-1} BB^T (V^{-1})^T (\Lambda^T)^k\Big] V^T = VMV^T. \end{aligned} \tag{3.57}$$

Let M_{ij} be the entry in the i-th row and j-th column of the matrix M, then:

$$M_{ij} = \Big[V^{-1}BB^T(V^{-1})^T\Big]_{ij} \sum_{k=0}^{\infty} (\gamma_i\gamma_j)^k = \Big[V^{-1}B\Big]_i \Big[V^{-1}B\Big]_j \frac{1}{1-\gamma_i\gamma_j}. \tag{3.58}$$

Note that the existence of the solution to the previous equation is granted by the fact that all the eigenvalues of A have modulus smaller than 1, hence $|\gamma_i\gamma_j| < 1$.
The Gramian can hence be written expanding matrix M as:

$$\mathcal{W} = V\text{diag}(V^{-1}B)\mathcal{C}\text{diag}(V^{-1}B)^T V^T, \qquad \mathcal{C}_{ij} = \frac{1}{1-\gamma_i\gamma_j}. \tag{3.59}$$

The columns of the matrix V are the normalized right eigenvectors of A, whereas the rows of V^{-1} are the normalized left eigenvectors and transpose of A hence:

$$V = \frac{1}{\alpha}\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \qquad V^{-1} = \frac{1}{\alpha}\begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix} \tag{3.60}$$

All the eigenvectors have the same norm α which is equal to:

$$\alpha = \sqrt{\sum_{k=1}^{n} \sin\left(\frac{k\pi}{n+1}\right)^2} = \sqrt{\frac{n+1}{2}} \tag{3.61}$$

Moreover since we are considering $B = \begin{bmatrix} b_u & \mathbf{0} \end{bmatrix}$:

$$D = \mathrm{diag}(V^{-1}B) = \mathrm{diag}(\frac{1}{\alpha}\begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix}\begin{bmatrix} b_u \\ \vdots \\ 0 \end{bmatrix}) = \frac{b_u}{\alpha}\begin{bmatrix} y_{11} & \dots & \dots & \dots \\ \dots & y_{21} & \ddots & \dots \\ \dots & \dots & \ddots & \dots \\ \dots & \dots & \dots & y_{n1} \end{bmatrix} = \mathrm{diag}(V^{-1}B)^T. \tag{3.62}$$

The matrix $\mathcal{C}$ can be equivalently rewritten as

$$\mathcal{C}_{ij} = \frac{1}{1-\gamma_i\gamma_j} = \frac{1}{\gamma_i}\frac{1}{\frac{1}{\gamma_i}-\gamma_j} = \frac{\bar{x}_i}{\bar{x}_i+\bar{y}_j} \qquad \text{with} \quad \begin{cases} \bar{x}_i = \frac{1}{\gamma_i}, \\ \bar{y}_j = -\gamma_j. \end{cases} \tag{3.63}$$

This last definition of the entries of the matrix $\mathcal{C}$ is well-posed since we have set $\lambda \neq 0$, hence no eigenvalue is in 0 and the quantity $\frac{1}{\gamma_i}$ is always finite.
Let $\bar{x} = \begin{bmatrix} \bar{x}_1 & \dots & \bar{x}_n \end{bmatrix}$, exploiting the definition in the previous equation the matrix $\mathcal{C}$ can be written as $\mathcal{C} = \Gamma\bar{\mathcal{C}}$ where $\Gamma = \mathrm{diag}(\bar{x})$ and $\bar{\mathcal{C}}$ is a Cauchy matrix with entries:

$$\bar{\mathcal{C}}_{ij} = \frac{1}{\bar{x}_i+\bar{y}_j}. \tag{3.64}$$

For Cauchy matrices there exist an explicit formula to compute the inverse which is given entry wise by:

$$[\bar{\mathcal{C}}]^{-1}_{ij} = (\bar{y}_i+\bar{x}_j)\frac{\prod_{k\neq i}(\bar{x}_j+\bar{y}_k)\prod_{k\neq j}(\bar{y}_i+\bar{x}_k)}{\prod_{k\neq j}(\bar{x}_j-\bar{x}_k)\prod_{k\neq i}(\bar{y}_i-\bar{y}_k)} \tag{3.65}$$

Since we would like to compute the trace of the inverse of the Gramian we are mainly interested in the diagonal entries of the matrix $\bar{\mathcal{C}}$:

$$\begin{aligned} [\bar{\mathcal{C}}]^{-1}_{ii} &= (\bar{y}_i+\bar{x}_i)\frac{\prod_{k\neq i}(\bar{x}_i+\bar{y}_k)\prod_{k\neq i}(\bar{y}_i+\bar{x}_k)}{\prod_{k\neq i}(\bar{x}_i-\bar{x}_k)\prod_{k\neq i}(\bar{y}_i-\bar{y}_k)} = (\bar{y}_i+\bar{x}_i)\frac{\prod_{k\neq i}(\bar{x}_i+\bar{y}_k)(\bar{y}_i+\bar{x}_k)}{\prod_{k\neq i}(\bar{x}_i-\bar{x}_k)(\bar{y}_i-\bar{y}_k)} = \\ &= (\frac{1}{\gamma_i}-\gamma_i)\prod_{k\neq i}\frac{\frac{1}{\gamma_i}-\gamma_k}{\frac{1}{\gamma_i}-\frac{1}{\gamma_k}}\frac{\frac{1}{\gamma_k}-\gamma_i}{\gamma_k-\gamma_i} = (\frac{1-\gamma_i\gamma_i}{\gamma_i})\prod_{k\neq i}\frac{1-\gamma_i\gamma_k}{\gamma_i\frac{\gamma_k-\gamma_i}{\gamma_i\gamma_k}}\frac{1-\gamma_i\gamma_k}{(\gamma_k-\gamma_i)\gamma_k} = \\ &= \frac{(1-\gamma_i^2)}{\gamma_i}\prod_{k\neq i}\left|\frac{1-\gamma_i\gamma_k}{\gamma_k-\gamma_i}\right|^2. \end{aligned} \tag{3.66}$$

Thanks to the introduced Cauchy matrix we can hence compute explicitly the trace of the inverse of the Gramian:

$$\begin{aligned} \mathrm{trace}[W^{-1}] &= \mathrm{trace}[V^{-T}D^{-T}\bar{\mathcal{C}}^{-1}\Gamma^{-1}D^{-1}V^{-1}] = \mathrm{trace}[D^{-T}\bar{\mathcal{C}}^{-1}\Gamma^{-1}D^{-1}V^{-1}V^{-T}] = \\ &= \mathrm{trace}[D^{-T}\bar{\mathcal{C}}^{-1}\Gamma^{-1}D^{-1}] = \sum_{i=1}^{n}\frac{\gamma_i(1-\gamma_i^2)}{d_{ii}^2\gamma_i}\prod_{k\neq i}\left|\frac{1-\gamma_i\gamma_k}{\gamma_k-\gamma_i}\right|^2 = \\ &= \sum_{i=1}^{n}\frac{(1-\gamma_i^2)}{d_{ii}^2}\prod_{k\neq i}\left|\frac{1-\gamma_i\gamma_k}{\gamma_k-\gamma_i}\right|^2. \end{aligned} \tag{3.67}$$

■

As a by-product of the previous proposition and proof it is possible to find an analytic expression for the Gramian. This provides us with an alternative method to compute it without resorting to the discrete Lyapunov equations. This expression is useful in numerical simulations to compute the Gramian with higher precision with respect to Lyapunov equations.

Corollary 1 *Consider the symmetric line network described in equation* (3.14) *with single input on the first node and positive self loops. Let the matrix A of the system be stable. The entries of the controllability Gramian for the simple network read:*

$$W_{ij} = \frac{\sum_k^n \sum_s^n b_u^2 \sin(\frac{si\pi}{n+1}) \sin(\frac{kj\pi}{n+1}) \sin(\frac{s\pi}{n+1}) \sin(\frac{k\pi}{n+1})}{\frac{n+1}{2b_u^2}(1-\lambda^2 - 2|b|\lambda\cos(\frac{k\pi}{n+1}) - 2|b|\lambda\cos(\frac{s\pi}{n+1}) - 4b^2\cos(\frac{s\pi}{n+1})\cos(\frac{k\pi}{n+1}))} \tag{3.68}$$

Proof: By following the same reasoning and with the same matrix definition of previous proof, the controllability Gramian can be written as;

$$\mathcal{W} = V \text{diag}(V^{-1}B)\, \mathcal{C}\, \text{diag}(V^{-1}B)^T V^T, \qquad \mathcal{C}_{ij} = \frac{1}{1-\gamma_i\gamma_j} \tag{3.69}$$

Let $\hat{\mathcal{C}} = D\mathcal{C}D^T$ then

$$\begin{aligned} \hat{\mathcal{C}}_{ij} = D_{ii}\mathcal{C}_{ij}D_{jj} &= \frac{1}{\alpha^2}\frac{y_{i1}y_{1j}}{1-\gamma_i\gamma_j} = \frac{1}{\alpha^2}\frac{\sin(\frac{i\pi}{n+1})\sin(\frac{j\pi}{n+1})}{1-(\lambda+2|b|\cos(\frac{i\pi}{n+1}))(\lambda+2|b|\cos(\frac{j\pi}{n+1}))} \\ &= \frac{1}{\alpha^2}\frac{\sin(\frac{i\pi}{n+1})\sin(\frac{j\pi}{n+1})}{(1-\lambda^2-2|b|\lambda\cos(\frac{i\pi}{n+1})-2|b|\lambda\cos(\frac{j\pi}{n+1})-4b^2\cos(\frac{i\pi}{n+1})\cos(\frac{j\pi}{n+1}))}. \end{aligned} \tag{3.70}$$

Finally $\mathcal{W} = V\hat{\mathcal{C}}V^T$, hence:

$$\mathcal{W}_{ij} = \frac{\sum_k^n \sum_s^n b_u^2 \sin(\frac{si\pi}{n+1}) \sin(\frac{kj\pi}{n+1}) \sin(\frac{s\pi}{n+1}) \sin(\frac{k\pi}{n+1})}{\frac{n+1}{2b_u^2}(1-\lambda^2 - 2|b|\lambda\cos(\frac{k\pi}{n+1}) - 2|b|\lambda\cos(\frac{s\pi}{n+1}) - 4b^2\cos(\frac{s\pi}{n+1})\cos(\frac{k\pi}{n+1}))} \tag{3.71}$$

■

In order to prove that the lifted dynamics bring us advantages it is sufficient to prove that some advantages exists in a specific configuration. We hence decide to compute the Gramian for the expanded network with $\lambda = 0$ and $a = 0$ as it is the unique configuration which always satisfies the condition on the modulus of the eigenvalues for any choice of self-loop of the simple network.

Remark 10 *From a structural controllability perspective, it would not be a natural choice to decouple the left-oriented line and right-oriented line by setting $\hat{a} = 0$. The input enters the lifted network on both the nodes of the first subsystem, however while it propagates through the right-oriented line, it can not be propagated through the left-oriented line since $(1,2) \notin \hat{\mathcal{E}}$. However the task is to control the output of the system intended as the induced state $x_I(t)$ and not the state of the lifted network, so we can let the left-oriented line act as a drift and control only the right oriented line. The validity of this approach is proved below, where we find the expression of the Gramian for a line network with $\hat{a} = 0, \hat{b} \neq 0, \hat{\lambda} = 0$, it is always positive definite, hence $x_I(t)$ is controllable.*

Proposition 4 *Consider the symmetric lifted network described by equation* (3.24) *with $\hat{a} = 0$, $\hat{\lambda} = 0$ then:*

$$\text{trace}[(\mathcal{W}_e^O)^{-1}] = \frac{1}{4b_u^2} + \sum_{i=1}^{2n-2} \frac{1}{b^i b_u^2}. \tag{3.72}$$

Proof: The matrix A_e is block triangular with $A_{e,lr} = 0 = A_{e,rl}$. The block $A_{e,ll}$ is upper triangular with all the elements on the diagonal equal to zero, all the eigenvalues of $A_{e,ll}$ are hence equal to 0. The block $A_{e,rl}$ is lower triangular with all the elements on the diagonal equal to zero, similarly as before all the eigenvalues of $A_{e,ll}$ are hence equal to 0.
Since $A_{e,lr} = 0 = A_{e,rl}$ the eigenvalues of the matrix A_e are the union of the eigenvalues of the blocks on the diagonal and since both the eigenvalues of $A_{e,ll}$ and $A_{e,rr}$ are all equal to zero, so are the eigenvalues of A_e. We have hence proved the matrix to be nilpotent.
It is possible to prove that A_e can be brought to its Jordan form by means of a permutation matrix T where the n-th column is the canonical vector e_n and the $n+1$ column is e_{n+1}; the column i is $e_i \cdot \hat{b}^{n-i}$ for $i \in 1, \dots, n-1$ and the column $n+j$ is $e_{n-j+1} \cdot \hat{b}^{n-j}$ for $j \in \{1, \dots, n-1\}$.
For sake of completeness we write below an example of permutation matrix for a lifted network with 3 subsystems

$$T = \begin{bmatrix} b^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & b & 0 \\ 0 & 0 & 0 & b^2 & 0 & 0 \end{bmatrix} \tag{3.73}$$

By looking to the Jordan form of A_e there are 2 Jordan blocks of dimension n, hence the minimum nilpotency index is exactly n. Since the matrix A_e is nilpotent with nilpotency index n the infinite-horizon controllability Gramian is equal to the n step controllability Gramian. In fact considering the expression:

$$\mathcal{W}_e = \sum_{k=0}^{\infty} A_e^k B_e B_e^T (A_e^k)^T \tag{3.74}$$

each term employing A^k with $k > n$ is 0 as $A^k = 0$.
To compute the Gramian, we first compute the n-step controllability matrix:

$$\mathcal{C}_n = \begin{bmatrix} B & A_e B & \dots & A_e^{n-1} B \end{bmatrix} \tag{3.75}$$

where:

$$B = \begin{bmatrix} \hat{b}_u \\ 0 \\ 0 \\ \vdots \\ 0 \\ \hat{b}_u \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad A_e B = \begin{bmatrix} \hat{b}_u \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \hat{b}\hat{b}_u \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad A_e A_e B = \begin{bmatrix} \hat{b}_u \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \hat{b}^2 \hat{b}_u \\ \vdots \\ 0 \end{bmatrix}, \quad \dots. \tag{3.76}$$

The n-step controllability Gramian is given by:

$$\mathcal{C}_n \mathcal{C}_n^T = \begin{bmatrix} \hat{b}_u^2 & 0 & 0 & \dots & 0 & \hat{b}_u^2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{b}_u^2 & 0 & 0 & \dots & 0 & \hat{b}_u^2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & b\hat{b}_u^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & \hat{b}^{2(n-1)} \hat{b}_u^2 \end{bmatrix}. \tag{3.77}$$

The output controllability Gramian is then given by:

$$\mathcal{W}_e^O = \tilde{C}\mathcal{C}_n\mathcal{C}_n^T\tilde{C}^T = \begin{bmatrix} 4\hat{b}_u^2 & 0 & 0 & \dots & 0 \\ 0 & \hat{b}^2\hat{b}_u^2 & 0 & \dots & 0 \\ 0 & 0 & \hat{b}^4\hat{b}_u^2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \hat{b}^{2(n-1)}\hat{b}_u^2 \end{bmatrix}. \tag{3.78}$$

Hence by taking the inverse of the Gramian which can be easily computed by replacing each of the scalar entries on the diagonal with its inverse, the required metric is obtained by summing the entries on the diagonal:

$$\text{trace}[(\mathcal{W}_e^O)^{-1}] = \frac{1}{4\hat{b}_u^2} + \sum_{i=1}^{2n-2} \frac{1}{\hat{b}^i\hat{b}_u^2}. \tag{3.79}$$

■

3.4 Computation of the Asymptotic rate

As already discussed we are interested in understating the advantages in terms of control energy the lifted configuration bring to us with respect to the simple network as the sizes of the two networks grow. In this section we consider again the trace of the Gramian and we compute the **asymptotic rate** which is defined as the exponential rate of growth of the average control energy metric as the size of the network tends to infinity, i.e.,

$$\rho = \lim_{n\to\infty} \frac{1}{n} \log(\text{trace}(\mathcal{W}^{-1})) \tag{3.80}$$

We will compute the asymptotic rates both for the simple network and the expanded network and compare them in order to prove the advantages of the lifted configuration. We start by computing the asymptotic rate for a specific lifted network configuration of interest.

Proposition 5 *Consider the symmetric lifted network described by equation* (3.24) *with* $\hat{a} = 0$, $\hat{\lambda} = 0$. *Then the asymptotic rate is given by:*

$$\rho_e = 2\log\left(\frac{1}{|\hat{b}|}\right). \tag{3.81}$$

Proof: It is sufficient to compute the limit defined in equation (3.80) and use the expression of the Gramian in (3.72). Namely,

$$\rho_e = \lim_{n\to\infty} \frac{1}{n} \log(\text{trace}((\mathcal{W}_e^O)^{-1})) = \lim_{n\to\infty} \frac{1}{n} \log\left(\frac{1}{4\hat{b}_u^2} + \frac{1}{\hat{b}_u^2}\sum_{k=1}^{n-1}\frac{1}{\hat{b}^{2k}}\right).$$

Since $|b| < 1$ to satisfy the stability constraint, it holds

$$\log\left(\frac{1}{4\hat{b}_u^2} + \frac{1}{\hat{b}_u^2\hat{b}^{2(n-1)}}\right) \leq \log\left(\frac{1}{4\hat{b}_u^2} + \frac{1}{\hat{b}_u^2}\sum_{k=1}^{n-1}\frac{1}{\hat{b}^{2k}}\right) \leq \log\left(\frac{1}{4\hat{b}_u^2} + \frac{n-1}{\hat{b}_u^2\hat{b}^{2(n-1)}}\right).$$

From the latter inequalities it follows that for large n

$$\frac{1}{n}\log\left(\frac{1}{4\hat{b}_u^2} + \frac{1}{\hat{b}_u^2}\sum_{k=1}^{n-1}\frac{1}{\hat{b}^{2k}}\right) \approx \frac{1}{n}\log\left(\frac{1}{\hat{b}^{2(n-1)}}\right) \approx 2\log\left(\frac{1}{|\hat{b}|}\right),$$

which yields the thesis. ■

As it concerns the computation of the asymptotic rate for the simple line network things gets more complicated. A first attempt to find a bound on the asymptotic rate exploited the explicit formula of the inverse of the Gramian found in Proposition 3, however this did not lead us to any interesting conclusion.

An alternative path exploits the asymptotic density for the eigenvalues of the symmetric line network fed with a single input.

In [7] it is shown how for n large the eigenvalues of matrix A of the symmetric network described by equation (3.14) by choosing $b_u = 1$ take values in the interval $[\lambda - 2|b|, \lambda + 2|b|]$ with density:

$$\phi(\mu) = \frac{1}{\pi\sqrt{4b^2 - (\mu-\lambda)^2}}\mathbf{1}_{[\lambda-2|b|,\lambda+2|b|]}(\mu). \tag{3.82}$$

In the literature the above function is called **arcsine** density.

Remark 11 *Thanks to the introduced density it is possible to notice how a constrain on the stability for the matrix A once fixed $\lambda > 0$ as n grows to infinity is*

$$\frac{(\lambda-1)}{2} < b < \frac{(1-\lambda)}{2}. \tag{3.83}$$

By adapting the reasoning followed in [8] to the discrete-time case, it is possible to prove that:

$$\lim_{n\to\infty}\frac{1}{n}\log(\mathrm{trace}(\mathcal{W}^{-1})) = 2\max_{\gamma\in[\lambda-2|b|,\lambda+2|b|]}\int_{\lambda-2|b|}^{\lambda+2|b|}\log\Big|\frac{1-\mu\gamma}{\gamma-\mu}\Big|\frac{1}{\pi\sqrt{4b^2-(\mu-\lambda)^2}}d\mu \tag{3.84}$$

The following proposition gives the explicit analytical expression of the asymptotic rate by solving the above integral.

Proposition 6 *Consider the symmetric line network described in equation* (3.14) *with single input on the first node and non negative self loops. Let the matrix A of the system be stable, then:*

$$\rho = \max_{\gamma\in[\lambda-2|b|,\lambda+2|b|]} 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - 2\log|b|. \tag{3.85}$$

Proof: We consider the formula for the asymptotic rate in equation (3.84). Let:

$$\begin{aligned}
f(\gamma) &= 2\int_{\lambda-2|b|}^{\lambda+2|b|}\log\Big|\frac{1-\mu\gamma}{\gamma-\mu}\Big|\frac{1}{\pi\sqrt{4b^2-(\mu-\lambda)^2}}d\mu = 2\int_{\lambda-2|b|}^{\lambda+2|b|}\log\Big|\frac{1-\mu\gamma}{\gamma-\mu}\Big|p(\mu)d\mu \\
&= \int_{\lambda-2|b|}^{\lambda+2|b|}\log(1-\mu\gamma)^2p(\mu)d\mu - \int_{\lambda-2|b|}^{\lambda+2|b|}\log(1-\mu\gamma)^2p(\mu)d\mu \\
&= \int_{\lambda-2|b|}^{\lambda+2|b|}\log(\gamma^2)p(\mu)d\mu + \int_{\lambda-2|b|}^{\lambda+2|b|}\log((\frac{1}{\gamma}-\mu)^2)p(\mu)d\mu - \int_{\lambda-2|b|}^{\lambda+2|b|}\log((\gamma-\mu)^2)p(\mu)d\mu \\
&= \log(\gamma^2) + \mathbb{E}_p\Big[\log((\frac{1}{\gamma}-\mu)^2)\Big] - \mathbb{E}_p\Big[\log((\gamma-\mu)^2)\Big].
\end{aligned}$$

From [9] corollary 3 we know that given the density:

$$q(t) = \frac{1}{\pi\sqrt{(t-c)(c-d)}}, \quad c < t < d, \tag{3.86}$$

and ζ a random variable, then:

$$\mathbb{E}_q\log(\zeta - z)^2 = \begin{cases} 2\log(d-c) - 4\log(2) & \text{if } c \le z \le d \\ 2\log(d-c) + 2\log(|x_z| + \sqrt{x_z^2-1}) - 4\log(2) & \text{if } z < c \text{ or } z > d, \end{cases} \tag{3.87}$$

where $x_z = -1 + \frac{2(z-c)}{d-c}$.

The density p equals q by setting $c = \lambda - 2|b|$ and $d = \lambda + 2|b|$, then by the previous equation since μ has density p and $\lambda - 2|b| \leq \mu \leq \lambda + 2|b|$:

$$\mathbb{E}_p\Big[\log((\gamma-\mu)^2)\Big] = \mathbb{E}_p\Big[\log((\mu-\gamma)^2)\Big] = 2\log(\lambda + 2|b| - \lambda + 2|b|) - 4\log(2) = \\ = 2\log(4|b|) - 4\log(2) = 2\log(4|b|/4) = 2\log(|b|).$$

Now we consider the quantity:

$$\mathbb{E}_p\Big[\log((\frac{1}{\gamma}-\mu)^2)\Big] = \mathbb{E}_p\Big[\log((\mu-\frac{1}{\gamma})^2)\Big]. \tag{3.88}$$

Since $\frac{1}{\gamma}$ is such that $\frac{1}{\gamma} > \lambda + 2|b|$ or $\frac{1}{\gamma} < \lambda - 2|b|$ then by exploiting the second equality in (3.87):

$$\begin{aligned} \mathbb{E}_p\Big[\log((\frac{1}{\gamma}-\mu)^2)\Big] &= 2\log(|b|) + 2\log\Big(\frac{|\frac{1}{\gamma}-\lambda| + \sqrt{(\frac{1}{\gamma}-\lambda)^2 - 4b^2}}{2b}\Big) \\ &= 2\log(|b|) + 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2b|\gamma|}\Big) \\ &= 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - 2\log(|\gamma|) \\ &= 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - \log(\gamma^2). \end{aligned}$$

Hence, exploiting the computed quantities:

$$\begin{aligned} f(\gamma) &= \log(\gamma^2) + \mathbb{E}_p\Big[\log((\frac{1}{\gamma}-\mu)^2)\Big] - \mathbb{E}_p\Big[\log((\gamma-\mu)^2)\Big] = \\ &= \log(\gamma^2) + 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - \log(\gamma^2) - 2\log(|b|) \\ &= 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - 2\log(|b|). \end{aligned}$$

■

Once analytic expressions for the **asymptotic rates** of the lifted and simple systems have been derived, it is possible to compare these two quantities.

Proposition 7 *Consider the symmetric line network with positive self loops described in equation* (3.14) *and the symmetric lifted line network of equation* (3.24), *where we set* $\hat{b} = b$, $\hat{b}_u = \frac{b_u}{2}$. *Let* $|\lambda \pm 2b| < 1, \gamma \in [\lambda - 2|b|, \lambda + 2|b|]$ *then:*

$$\rho > \rho_e \iff \exists\,\lambda,\, b,\, \gamma \;\; s.t. \;\; |1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2} > 2. \tag{3.89}$$

Proof: It is sufficient to exploit the two expressions for the asymptotic gains identified in propositions 5 and 6:

$$\begin{aligned} &\rho > \rho_e \\ &\iff \max_{\gamma\in[\lambda-2|b|,\lambda+2|b|]} 2\log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) - 2\log|b| > -2\log(|b|) \\ &\iff \exists\lambda, b, \gamma \quad \text{s.t.} \quad \log\Big(\frac{|1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2}}{2}\Big) > 0 \\ &\iff |1-\lambda\gamma| + \sqrt{(1-\lambda\gamma)^2 - 4b^2\gamma^2} > 2. \end{aligned} \tag{3.90}$$

■

We now provide a couple of examples in which the given equality can be used to prove advantages/disadvantages of the lifted configuration with respect to the simple one.

Example 6 (Advantages)
Consider the lifted and simple symmetric line networks with parameters: $\hat{a} = 0, \hat{\lambda} = 0, b_u = 1, \hat{b}_u = 0.5, \lambda = 0.2, b = 0.39.$
The stability requirement are satisfied by:

$$\begin{aligned} |\lambda - 2b| = |0.2 - 0.78| = 0.58 < 1, \\ |\lambda + 2b| = |0.2 + 0.78| = 0.98 < 1. \end{aligned} \tag{3.91}$$

The constraints on the non amplifying behaviour are granted by the choice of parameters $\hat{b}_u = \frac{b_u}{2} = 0.5,\ \hat{b} = b = 0.39,\ \lambda = 0.2 > 0$ *and* $|\hat{\lambda}| < |\lambda| = 0.2.$
We set $\gamma \in [\lambda - 2|b|, \lambda + 2|b|] = [-0.6, 1].$
We choose $\gamma = -0.6$ *and we exploit the inequality in the previous proposition to check the asymptotic advantage of this lifted configuration:*

$$|1 - \lambda\gamma| + \sqrt{(1 - \lambda\gamma)^2 - 4b^2\gamma^2} = 1.12 + \sqrt{(1.12)^2 - 0.22} \approx 2.27 > 2. \tag{3.92}$$

Hence we proved that in this specific scenario there are advantages in using the proposed lifted configuration.

Example 7 (No advantages)
Consider the lifted and simple symmetric line networks with parameters: $\hat{a} = 0, \hat{\lambda} = 0, \lambda = 0.2, b = 0.05,\ b_u = 1, \hat{b}_u = 0.5.$
The stability requirement are satisfied by:

$$\begin{aligned} |\lambda - 2b| = |0.2 - 0.1| = 0.1 < 1, \\ |\lambda + 2b| = |0.2 + 0.1| = 0.3 < 1. \end{aligned} \tag{3.93}$$

The constraints on the non amplifying behaviour are granted by the choice of parameters $\hat{b}_u = \frac{b_u}{2} = 0.5,\ \hat{b} = b = 0.05,\ \lambda = 0.2 > 0$ *and* $|\hat{\lambda}| < |\lambda| = 0.2.$
We set $\gamma \in [\lambda - 2|b|, \lambda + 2|b|] = [0.1, 0.3].$
We choose γ *equal to its maximum which can be found to be* $\gamma = 0.1$ *and we compute the inequality:*

$$|1 - \lambda\gamma| + \sqrt{(1 - \lambda\gamma)^2 - 4b^2\gamma^2} \approx 1.9599 < 2. \tag{3.94}$$

In this configuration there are no advantages in using the proposed lifted configuration.

Remark 12 *If we choose a simple network with* λ, b *such that* $0 \in [\lambda - 2|b|, \lambda + 2|b|]$ *we can set* $\gamma = 0$*, the left hand side of the inequality of the previous proposition reads:*

$$|1 - \lambda\gamma| + \sqrt{(1 - \lambda\gamma)^2 - 4b^2\gamma^2} = 1 + 1 = 2 \tag{3.95}$$

The inequality is not satisfied however we achieve the same asymptotic rate of the simple network, hence by employing the proposed lifted configuration whenever there are no advantages at least we achieve the same performance of the non-lifted network.

Remark 13 *Thanks to the explicit expression for the asymptotic rate of the symmetric network we can show that if we have two symmetric networks with the same edge weights all equal to* $b \neq 0$ *but in the first network we set the self-loop* $\lambda_1 = 0$ *and in the second one we set* $\lambda_2 > 0$ *such that* $0 \in [\lambda_2 + 2|b|, \lambda_2 - 2|b|]$ *then the asymptotic rate of the first network is always smaller or equal to the asymptotic rate of the second network. We formally prove this below.*
Let $\gamma_2 \in [\lambda_2 - 2|b|, \lambda_2 + 2|b|]$, $\gamma_1 \in [-2|b|, 2|b|]$. *Let* ρ_1 *be the asymptotic rate of the first network and* ρ_2 *the asymptotic rate of the second network we solve the inequality* $\rho_2 \geq \rho_1$*:*

$$\begin{aligned}
&\rho_2 \geq \rho_1 \iff \\
&\max_{\gamma_2}\left[|1-\lambda_2\gamma| + \sqrt{(1-\lambda_2\gamma_2)^2 - 4b^2\gamma_2^2}\right] \geq \max_{\gamma_1}\left[|1-\lambda_1\gamma| + \sqrt{(1-\lambda_1\gamma_1)^2 - 4b^2\gamma_1^2}\right] \\
&\iff \max_{\gamma_2}\left[|1-\lambda_2\gamma_2| + \sqrt{(1-\lambda_2\gamma_2)^2 - 4b^2\gamma_2^2}\right] \geq \max_{\gamma_1}\left[1 + \sqrt{1 - 4b^2\gamma_1^2}\right] \geq 2
\end{aligned}$$

Since 0 belongs to the allowed interval of values for γ_2 *then we can set* γ_2 *equal to 0 and it follows:*

$$\rho_2 \geq |1 - \lambda_2 \cdot 0| + \sqrt{(1-\lambda_2 \cdot 0)^2 - 4b^2 \cdot 0^2} = 2 \geq 2. \tag{3.96}$$

To sum up, in this section we have provided the reader with an inequality which can be used to check whether there are advantages in terms of control energy in using the lifted configuration. Thanks to the two examples we know that there are cases in which the advantage exists, however we do not have advantages for all the possible choices of parameters in the simple network. Remark 12 highlights the convenience of employing the lifted configuration: without checking the inequality we know that if we modify the internal structure of each node of the network according to the lifted dynamics we may have advantages or at least we perform the same as the non-lifted network.

3.5 Simulation and numerical results

A set of numerical calculations and simulations have been performed in order to develop the intuition on the possible behaviors of the metrics and then to test the analytical results we have obtained. These simulations are reported in the current section.

In the following simulations we will restrict to the case of symmetric lifted and simple networks described in equation (3.24) and (3.14), respectively. Moreover we will always consider the edges of the two networks to have the same weights, i.e., $b = \hat{b}$. All the simulations have been performed using the software MATLAB R2023a, and the code developed for the main simulations can be found in the Appendix B.
We start now by describing the main ideas and procedures that have been used to develop the simulations. We will consider the two metrics:

$$\operatorname{trace}(\mathcal{W}^{-1}), \qquad \lambda_{\min}(\mathcal{W}). \tag{3.97}$$

Recall that the first metric is related to the average control energy over unit-norm random target states, whereas the second one to the worst-case control energy. In particular in the following of this section we will compare these metrics computed on the controllability Gramian $\mathcal{W}$ for the simple network and on the output controllability Gramian $\mathcal{W}_e^O$ for the expanded network, as described in previous sections. Since we are interested in assessing the behaviour of the metrics as the number of nodes grows we will often plot them in function of n, where n is the number of nodes in the simple network. In order to compute the controllability Gramian and output controllability Gramian we will let Matlab solve discrete-time Lyapunov equations. As an alternative we would have used the formulas developed in the previous sections, however the expressions for the lifted network have been computed only in the specific configuration $\hat{\lambda} = 0, \hat{a} = 0$ whereas we aim to inspect via simulation other parameter choices. It is a well known

result that for symmetric networks the aforementioned controllability metrics scale exponentially with the growth in number of nodes provided that the cardinality of the control nodes set remains constant.
We start the section by verifying the aforementioned exponential behaviour of the metrics thanks to a first simulation.
Consider the lifted and simple networks with the parameters set as described in table 3.1.

simple network			**lifted network**			
λ	b	b_u	$\hat{\lambda}$	$\hat{a}$	$\hat{b}$	$\hat{b}_u$
0.2	0.39	1	0.2	0.01	0.39	0.5

Table 3.1: Network parameter choice

In figure 3.5 the logarithms of the proposed metrics are plotted in function of the number of nodes in the simple network.
On the left side of the picture it is possible to find the metrics value plot for the simple network, whereas on the right side are depicted the metrics values for the expanded network. The subplots on the upper part of the figure are realized by computing the metric trace of the inverse of the Gramian, whereas the plots on the lower part of the figure refer to the minimum eigenvalue of the Gramian.
It is possible to appreciate from the figures the exponential behaviour of the metrics (it appears linear in the plots since we are taking the logarithm). Another observation that can be made is that while the trace of the Gramian grows exponentially, the metric minimum eigenvalue decays exponentially.

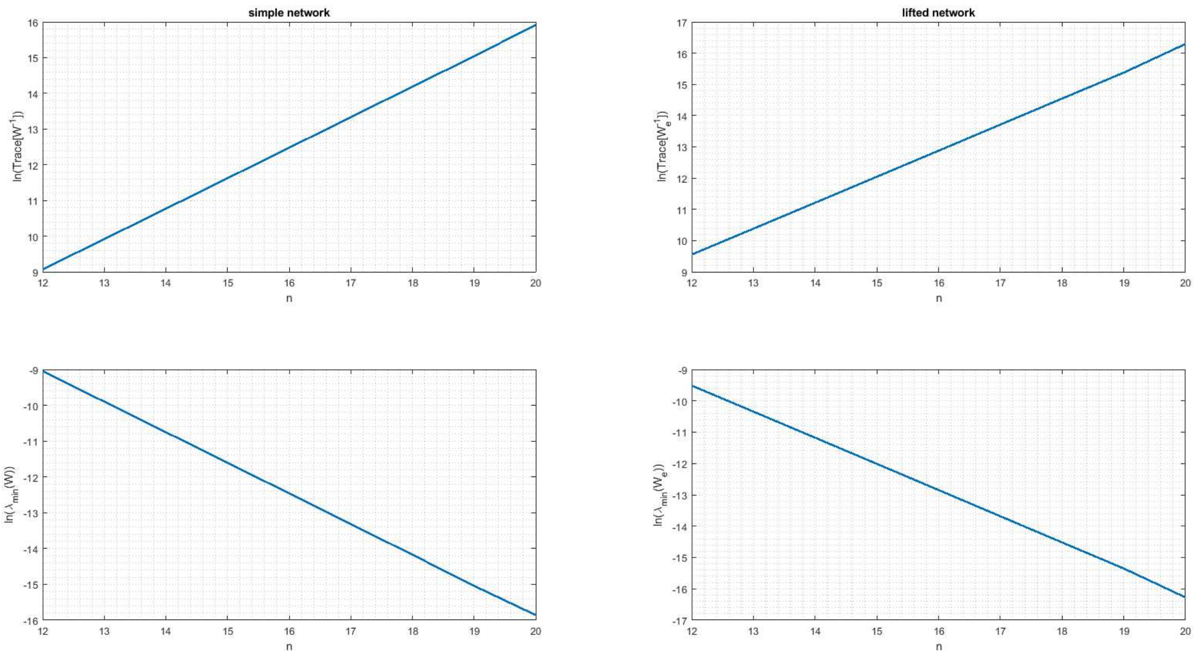

Figure 3.5: Shape of the metrics as the size of the network grows

We are now interested in comparing the asymptotic behaviour of the metrics for the lifted and simple line networks to verify the results suggested by the theory.
We have just shown how the metrics grow/decay exponentially with the number of nodes, we aim to compute the exponential factor of growth/decay, i.e., the asymptotic rate. If we apply the logarithm to the computed metrics the log-control energy shows a linear trend as depicted in the previous picture, the growth/decay factor is hence captured by the rate of growth/decay of the lines, which is their slope.
The simple and intuitive procedure we will follow in order to grasp from a simulation an estimate

of the asymptotic rate is based on the last reasoning.
We describe now the steps necessary to compute an estimate of the asymptotic rate:

1. We consider a first simple network with n_1 nodes and a second one with $n_2 > n_1$ nodes, we choose the same parameters λ and b for both the networks. Let $\mathcal{W}_{n_1}$ be the Gramian of the first network and $\mathcal{W}_{n_2}$ the Gramian of the second one. We compute the quantities:

$$\begin{aligned} e_{s,1} &= \log(\mathrm{trace}(\mathcal{W}_{n_2}^{-1})) - \log(\mathrm{trace}(\mathcal{W}_{n_1}^{-1})), \\ e_{s,2} &= \log(\lambda_{\min}(\mathcal{W}_{n_2})) - \log(\lambda_{\min}(\mathcal{W}_{n_1})). \end{aligned} \tag{3.98}$$

These quantities are related to the exponential growth/decay factors of the metrics for the simple network, intuitively they are connected to the slope of the lines in the previous picture.

2. We consider a lifted line network with $2n_1$ nodes and a second one with $2n_2$ nodes. Let $\mathcal{W}_{e,n_1}^O$ be the output controllability Gramian of the first network and and $\mathcal{W}_{e,n_2}^O$ the output controllability Gramian of the second one, we fix the same parameters $\hat{\lambda}$, $\hat{a}$, $\hat{b} = b$. We compute the quantities:

$$\begin{aligned} e_{e,1} &= \log(\mathrm{trace}((\mathcal{W}_{e,n_2}^O)^{-1})) - \log\mathrm{trace}(\mathcal{W}_{e,n_1}^O)^{-1})), \\ e_{e,2} &= \log(\lambda_{\min}(\mathcal{W}_{e,n_2}^O)) - \log(\lambda_{\min}(\mathcal{W}_{e,n_1}^O)). \end{aligned} \tag{3.99}$$

This quantities are related to the exponential growth/decay factors of the metrics for the lifted network.

3. Finally we compute the quantities:

$$\begin{aligned} e_1 &= e_{e,1} - e_{s,1}, \\ e_2 &= e_{e,2} - e_{s,2}. \end{aligned} \tag{3.100}$$

We are subtracting the estimate of the growth/decay rate of the simple line to the estimate of growth/decay rate of the lifted line.

The quantity e_1 give us an idea of the advantages we have in terms of control energy computed using the metric trace of the Gramian for the lifted network with respect to the simple one. Since the aforementioned metric grows with the cardinality of the network, in order to have advantages using the lift, we aim the asymptotic rate estimate for the simple line $e_{s,1}$ to be greater than the quantity $e_{e,1}$. If it holds, the control energy of the simple line grows faster than the one of the lifted line, which implies that as n grows large the lifted configuration requires less energy to be controlled than the simple one.
Therefore, the greater the negativity of e_1 the more advantage we have, if it is positive the simple network performs better.
The quantity e_2 instead quantifies the advantages we have in terms of control energy computed using the metric minimum eigenvalue of the Gramian. By following a reasoning similar as before, since the metric decays exponentially, the lifted configuration performs better if e_2 is positive.
While in the previous section we have presented analytically the proof of the advantages only for a specific configuration of the lifted network, i.e., the case where $\hat{\lambda} = 0$, $\hat{a} = 0$, we are now interested in assessing the advantages for different choices of parameters. To this aim we will fix the λ, b for the simple network and set a grid of possible choices of parameters for the lifted network. We will then compare the control energy of all the possible expanded networks built by setting the parameters according to the grid, with the control energy of the simple network. Since we will perform a grid search within the variables in a certain interval it is important to discard all the possible choices that does not respect the constraints which makes the comparison meaningful.
The constraints we will set are the following ones:

- **Constraints on the stability of the network**: The matrices A of the simple network and A_e of the expanded network needs to have all the eigenvalues inside the unit circle. If this constraint is not verified the infinite-horizon controllability Gramian does not exists.

- **Constraints on the controllability of the network**: The Gramian of the system needs to be positive definite in order for the system to be controllable.

- **Constraints on the allowed dynamics**: The parameters needs to be chosen in order to respect the constraints on the "non amplifying behaviour" of the lifted line network discussed in section 3.2.

- **Constraints on numerical errors**: Dealing with networks involves performing operations on large matrices, it becomes even more problematic when the matrices are close to be singular. In these cases Matlab is not always capable of performing operations such the inverse of a matrix (inverse of the Gramian). Some computational errors can be detected and the parameters which leads to these errors need to be discarded.

In all the simulations we will follow the procedure discussed above to compute the quantities e_1 and e_2 which allow us to compare the metrics for the lifted and simple systems. The two simple networks needed by the procedure are taken with $n_1 = 12$ and $n_2 = 16$ nodes, hence the corresponding lifted networks will have respectively $2n_1$=24 and $2n_2$=32 nodes.

We consider example 6, where we discussed thanks to the inequality in Proposition 7 the convenience of the lifted configuration with $\hat{\lambda} = 0$ and $\hat{a} = 0$. We aim to assess thanks to simulations if there are other choices of the parameters for the lifted network that lead us to an advantage. We set up a search grid by choosing the parameters as in table 3.2, where the notation c:j:d indicates that the parameters are chosen in the interval [c,d] with step j.

simple network			**lifted network**			
λ	b	b_u	$\hat{\lambda}$	$\hat{a}$	$\hat{b}$	$\hat{b_u}$
0.2	0.39	1	-1:0.01:1	-1:0.01:1	0.39	0.5

Table 3.2: Parameters choice for the first simulation

The results of the simulation are shown in figure 3.6. In the first subplot it is depicted the quantity e_1 in function of the choices of parameters $\hat{\lambda}$ and $\hat{a}$. In the second subplot the quantity e_2. The color bar reports the interval of the values taken by e_1 and e_2 for all the choices of parameters, all the values are negative for e_1 and positive for e_2. Hence the metric based on the trace grows faster for the simple network with respect to the lifted one, the metric minimum eigenvalue decays faster for the simple network. In particular the most advantageous configurations are the one achieved by setting $\lambda = 0$ and $a = \pm\lambda$. Intuitively this behaviour is justified by remark 13 where we proved that a line network with null self-loops performs better. By looking closely to the color-map however it is possible to notice that we have advantages also by setting $\hat{a} = 0$ and $\hat{\lambda} = \lambda$, hence by decoupling the right-oriented line and the left-oriented line and by choosing the same self-loops of the simple line.

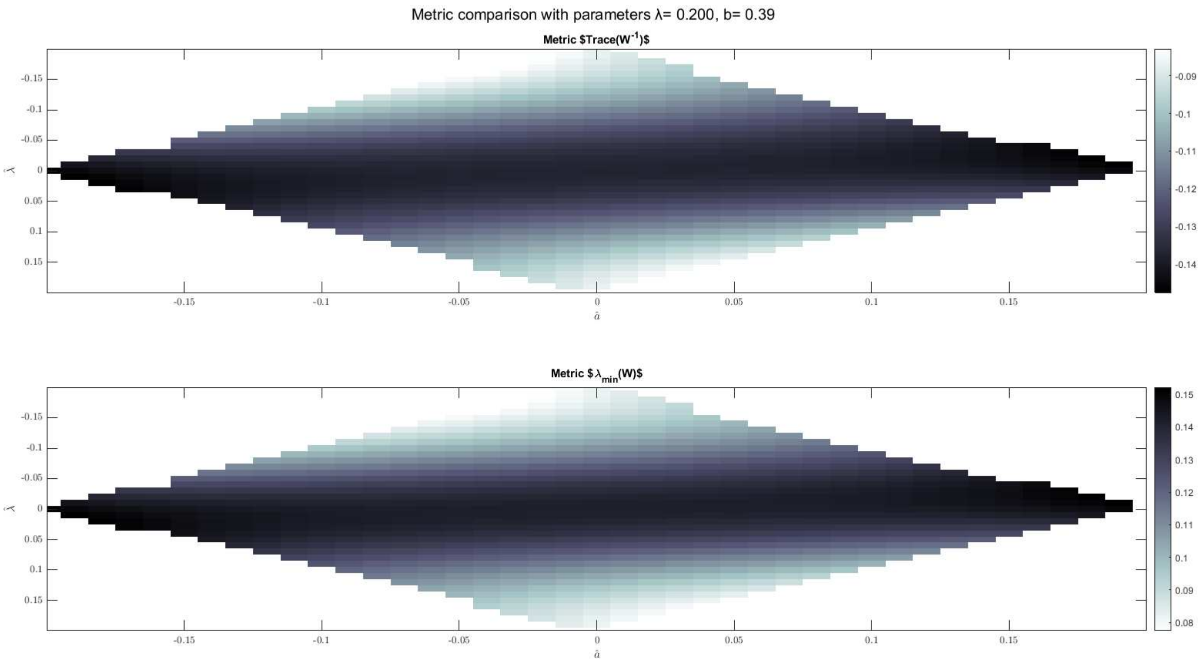

Figure 3.6: Results of the first simulation for the symmetric line

We consider now the second example in the previous section. We have already verified, thanks to the inequality of Proposition 7, there are no advantages in using the lifted configuration: now we aim to verify the result numerically. We set the parameters and the search grid according to table 3.3.

simple network			**lifted network**			
λ	b	b_u	$\hat{\lambda}$	$\hat{a}$	$\hat{b}$	$\hat{b}_u$
0.2	0.05	1	-1:0.01:1	-1:0.01:1	0.05	0.5

Table 3.3: Parameters choice for the second simulation

The result of the simulation is shown in figure 3.7, the values of e_1 are all positive whereas the values of e_2 are all negative, highlighting the fact that the simple network performs better than its lifted counterpart.

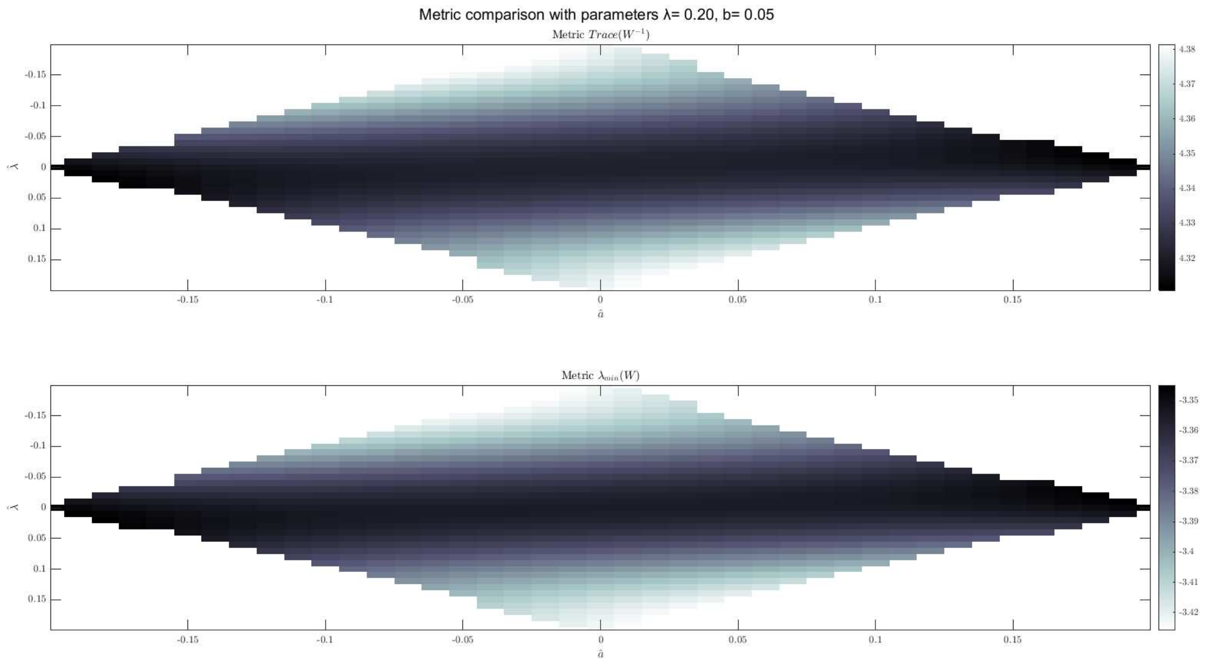

Figure 3.7: Results of the second simulation for the symmetric line

We end this section by providing a comparison between the metric trace of the inverse of the Gramian for the simple network computed using the analytic expression found in Proposition 3 and the solution computed by solving the discrete Lyapunov equations and by performing the inverse of the Gramian. The result is shown in figure 3.8, it is possible to appreciate the fact that while the discrete Lyapunov equation method starts to exhibit errors for networks with 22 nodes, if we exploit the analytical expression the result is still accurate for 30 nodes.

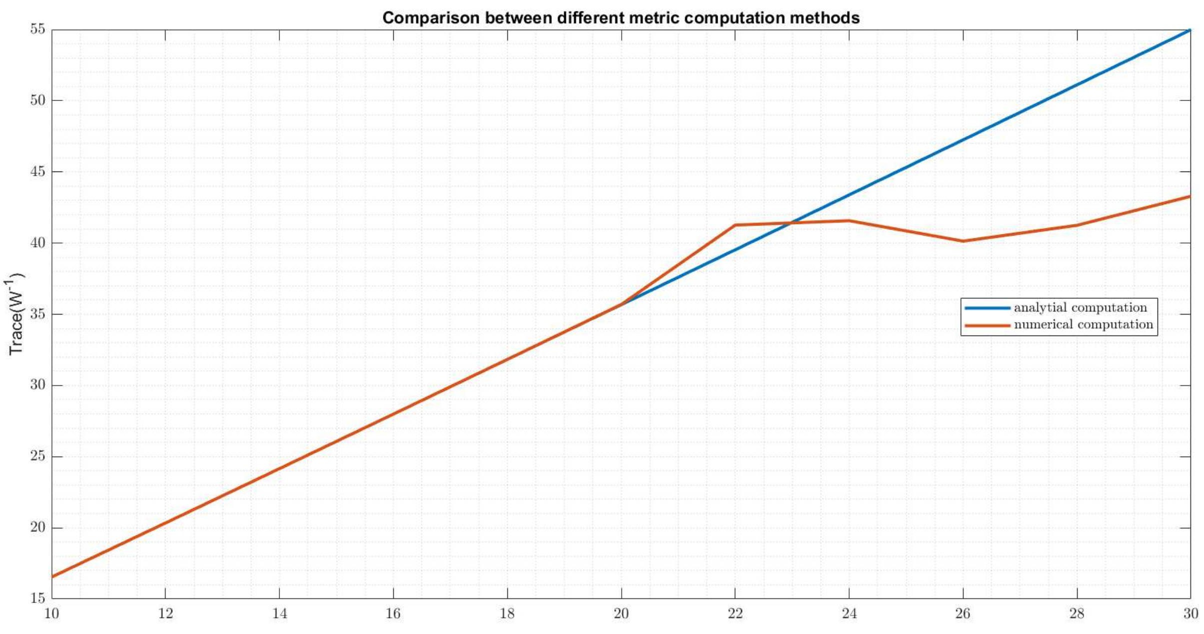

Figure 3.8: Comparison between different metric computation methods

Chapter 4

The Grid Network Case

4.1 Network description and control placement

In this chapter we apply the lift to an other well-known network topology: the grid network.

4.1.1 Non lifted network and control placement

In this section we describe the model of simple grid network we will use in the next section about numerical simulations.
We consider the directed grid network of n columns and M rows depicted in figure 4.1. To the network it is associated a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, nM\}$ is the set of nodes and $\mathcal{E}$ is the set of edges.
As usual the state of the network is described by the vector $x(t) \in \mathbb{R}^{nM}$ whose entry $x_i(t) \in \mathbb{R}$ is the state of node $i \in \mathcal{V}$ at time $t \in \mathbb{N}$, each node is hence considered to have scalar state.
We describe now in detail the topology of a general bidirectional grid network:

- **Each node**, except the first/last node of each row and the first/last node of each column, has four outgoing edges one connecting i with $i+1$, one connecting i with $i-1$ as for the line network, the weights assigned to these edges are respectively $b_{r,i+1}$ and $b_{l,i-1}$, moreover there are other two outgoing edges connecting i with $n+i$ and $i-n$, the weights assigned to these edges are respectively $b_{b,n+i}$ and $b_{t,n-i}$. There are also incoming edges from node $i+1$, $i-1$, $n+i$ and $n-i$ whose weights are labeled as $b_{l,i}$, $b_{r,i}$,$b_{t,i}$ and $b_{b,i}$.

- The **first node of each row** has three incoming/outgoing edges respectively from/to nodes $i+1$, $n+i$ and $n-i$.

- The **last node of each row** has three incoming/outgoing edges respectively from/to nodes $i-1$, $n+i$ and $n-i$.

- The **first node of each column** has three incoming/outgoing edges respectively from/to nodes $i+1$,$i-1$ and $n+i$.

- The **last node of each column** has three incoming/outgoing edges respectively from/to nodes $i-1$, $i+1$ and $n-i$.

- if a node is both the first/last of row and first/last of column it has only two outgoing/incoming edges.

- Each node $i \in \mathcal{V}$ has a **self loop** with weight λ_i.

- The weight associated to the input from the external environment to node i is labeled as $b_{u,i}$.

We now shift to describe the network as an interconnection of subsystems made of single nodes. The prototype single node system we will consider is depicted in figure 4.2.

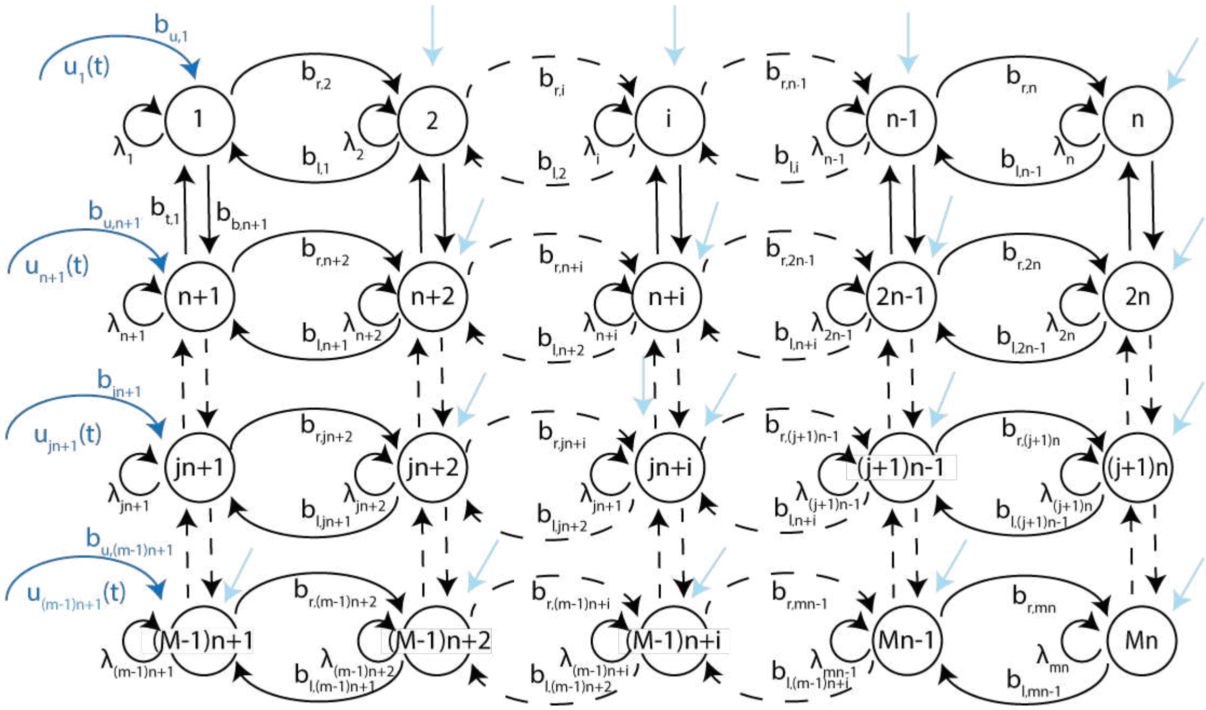

Figure 4.1: Simple grid network

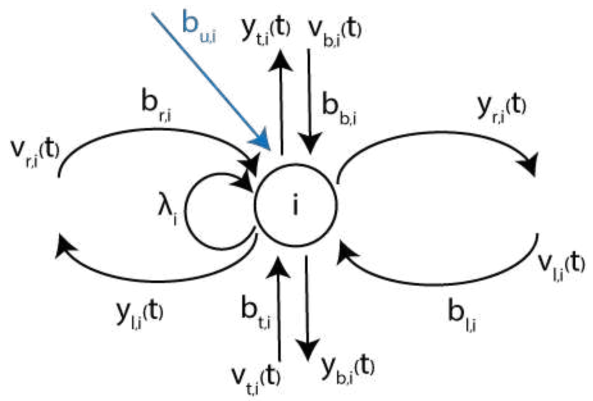

Figure 4.2: Simple grid subsystem

Each node is considered to have the same number of inputs and outputs as the number of incoming and outgoing edges described above. We label the inputs from nodes $i-1$, $i+1$, $n+i$, $n-i$ respectively as $v_{r,i}(t), v_{l,i}(t), v_{t,i}(t), v_{b,i}(t)$. We define as $v_i(t) = \begin{bmatrix} v_{r,i(t)} & v_{l,i}(t) & v_{t,i}(t) & v_{b,i}(t) \end{bmatrix}^T$ the vector grouping all the inputs from neighboring nodes. The weight associated to the edge $(j,i) \in \mathcal{E}$ is assigned to the corresponding input of node i (i.e., $v_{r,i}(t), v_{l,i}(t), v_{t,i}(t), v_{b,i}(t)$ affect the system with weights respectively $b_{r,i}$, $b_{l,i}$,$b_{t,i}$,$b_{b,i}$).
Let $y_i(t) = \begin{bmatrix} y_{r,i}(t) & y_{l,i}(t) & y_{t,i}(t) & y_{b,i}(t) \end{bmatrix}^T$ be the vector of the outputs whose entries represent respectively the outputs to nodes $i+1$, $i-1$, $n-i$, $n+i$, to each outgoing edge it is assigned as usual unit weight.
We label the external input as $u_{e,i}(t) \in \mathbb{R}$, it enters nodes i with weight $\hat{b}_{u,i}$.

According to the above description, the system can be written in state space form as:

$$\Sigma_i = \begin{cases} x_i(t+1) = A_i x_i(t) + B_{v,i} v_i(t) + B_{u,i} u_i(t) \\ y_i(t) = C_i x_i(t) \end{cases} \tag{4.1}$$

where the matrices describing the system are:

$$A_i = \lambda_i, \quad B_{v,i} = \begin{bmatrix} b_{r,i} & b_{l,i} & b_{t,i} & b_{b,i} \end{bmatrix}, \quad B_{u,i} = b_{u,i}, \quad C_{g,i} = \mathbb{1}_4. \tag{4.2}$$

The open loop transfer function of the single node system considering the state as output is:

$$W_i(z) = (z - A_i)^{-1} C_i = \frac{B_i}{s - \lambda_i} = \begin{bmatrix} \frac{b_{r,i}}{z-\lambda} & \frac{b_{l,i}}{z-\lambda} & \frac{b_{t,i}}{z-\lambda} & \frac{b_{b,i}}{z-\lambda} & \frac{b_u}{z-\lambda}, \end{bmatrix} \tag{4.3}$$

and the asymptotic gain is given by:

$$W_i(1) = \begin{bmatrix} \frac{b_{r,i}}{1-\lambda} & \frac{b_{l,i}}{1-\lambda} & \frac{b_{t,i}}{1-\lambda} & \frac{b_{b,i}}{1-\lambda} & \frac{b_u}{1-\lambda}. \end{bmatrix} \tag{4.4}$$

By imposing:

$$A_i = \lambda, \quad B_{v,i} = \begin{bmatrix} b_r & b_l & b_t & b_b \end{bmatrix}, \quad B_{u,i} = b_u, \quad C_i = \mathbb{1}_4, \quad \forall i \tag{4.5}$$

we assume all the systems to be identical and hence the matrices $A_i, B_{v,i}, B_{u,i}, C_i$ to be independent of i. Thanks to this assumption we can describe the overall network as the interconnection of single node systems.
By following the notation of equation (2.2) the state space form of the overall network system reads:

$$\Sigma : \begin{cases} x(t+1) = (I_{nM} \otimes A_i)x(t) + (I_{nM} \otimes B_{v,i})v(t) + B_u u(t) \\ y(t) = (I_{nM} \otimes C_i)x(t) \end{cases} \tag{4.6}$$

where:

$$B_u = \tilde{B}_u \otimes B_{u,i}, \qquad B_u \in \mathbb{R}^{n \times m}, \qquad v(t) = Ly(t). \tag{4.7}$$

It is possible to verify that the required L is

$$L = I_m \otimes S_{n,r} \otimes e_{2,4}e_{2,4}^T + I_m \otimes S_{n,l} \otimes e_{1,4}e_{1,4}^T + S_{m,r} \otimes I_n \otimes e_{4,4}e_{4,4}^T + S_{m,l} \otimes I_n \otimes e_{3,4}e_{3,4}^T, \tag{4.8}$$

where $S_{k,l}$ and $S_{k,r}$ are the usual shift matrices introduced in the previous chapter.
This time we choose to feed the network with a different input $u_i(t) \in \mathbb{R}^m$ for each row. The input enters the first node of each row.
The matrix $\tilde{B}_u$ should hence be selected in the following manner:

$$\tilde{B}_u = I_m \otimes e_{1,n}, \tag{4.9}$$

where $e_{1,n}$ is the first canonical vector of dimension n.

The overall state equation for the simple network can be written according to (2.4) as:

$$x(t+1) = \left[A_i \otimes I_n + (B_{v,i} \otimes I_n)L(C_i \otimes I_n)\right]x(t) + B_u u(t) = Ax(t) + B_u(t). \tag{4.10}$$

In the chapter about simulations we will consider a symmetric grid network, it can be obtained by setting the parameters $b_r = b_l = b$.
To better understand the structure of the matrix A we consider the following example:

Example 8 (2 × 3 Grid)
We consider a grid network with 3 columns and 2 rows. The matrices A and B of the system reads:

$$A = \begin{bmatrix} \lambda & b_l & 0 & b_t & 0 & 0 \\ b_r & \lambda & b_l & 0 & b_t & 0 \\ 0 & b_r & \lambda & 0 & 0 & b_t \\ b_b & 0 & 0 & \lambda & b_l & 0 \\ 0 & b_b & 0 & b_r & \lambda & b_l \\ 0 & 0 & b_b & 0 & b_r & \lambda \end{bmatrix}, \qquad B_u = \begin{bmatrix} b_u & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_u \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.11}$$

If we consider the case of symmetric grid network:

$$A = \begin{bmatrix} \lambda & b & 0 & b & 0 & 0 \\ b & \lambda & b & 0 & b & 0 \\ 0 & b & \lambda & 0 & 0 & b \\ b & 0 & 0 & \lambda & b & 0 \\ 0 & b & 0 & b & \lambda & b \\ 0 & 0 & b & 0 & b & \lambda \end{bmatrix}, \qquad B_u = \begin{bmatrix} b_u & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_u \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.12}$$

4.1.2 The lifted grid network

As done in the case of the line graph we build now the lifted network starting from the underlying graph. The way we will perform the lift is similar as before. In the line each node has two neighbors, hence in the associated lift we introduced two nodes for each vertex in the simple line in order to 'keep memory' of the direction from which the signal is received.
In the grid each node has at most 4 neighbors, hence the input can be received from 4 directions. By following the same idea as for the line we introduce a local memory to each node of the direction from which the input was received and process this information in order to better propagate the signal through the network and reduce the effort necessary to control the system. To this aim we will add 4 nodes in the lifted grid for each node in the simple grid, each of the nodes will then be connected to a different single input of the subsystem and a different single output. We are hence building:

- M right-oriented lines
- M left-oriented lines
- n top-oriented lines
- n bottom-oriented lines

We now discuss how the lift can be described by using the formalism introduced in chapter 2.

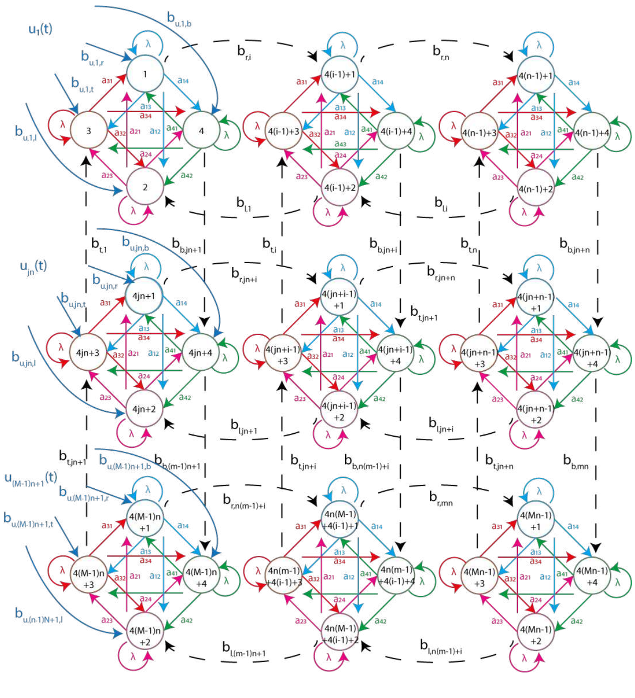

Figure 4.3: Lifted grid network

We consider the graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ describing the network in figure 3.3 where $\hat{\mathcal{V}}$ is the set of nodes and $\hat{\mathcal{E}}$ the set of edges. We design a map $\zeta : \hat{\mathcal{V}} \mapsto \mathcal{V}$ which associate to each node in $\mathcal{V}$ $k = 4$ nodes in $\hat{\mathcal{V}} = \{1, 2 \ldots, 4nM\}$. We reorder the nodes in the set $\hat{\mathcal{V}}$ so that $\zeta^{-1}(i) = \{(i-1)4+1, (i-1)4+2, (i-1)4+3, \ldots i4\}$.
We can hence think of the lifted network as composed by nM subsystems of 4 nodes, n subsystems for each column of the grid, M subsystems for each row.
The state of the network at time $t \in \mathbb{N}$ is described by the vector $x_e(t) = \begin{bmatrix} x_{e,1} & x_{e,2} & \ldots & x_{e,nM} \end{bmatrix} \in \mathbb{R}^{4nM}$ where $x_{e,i}(t) = \begin{bmatrix} x_{e,i,r}(t) & x_{e,i,l}(t) & x_{e,i,t}(t) & x_{e,i,b}(t) \end{bmatrix}$, $x_{e,i}(t) \in \mathbb{R}^4$, is the state of the i-th

subsystem, $i \in \{1, ..., n\}$. The prototype subsystem is shown in figure 4.4.
As done for the expanded line network we label the external input to subsystem i as $u_{e,i}(t) \in \mathbb{R}$ and we consider the same input to enter all the nodes of the subsystem, respectively $(i-1)4+1, (i-1)4+2, (i-1)4+3, (i-1)4+4$ with weights respectively $\hat{b}_{u,i,r}$, $\hat{b}_{u,i,l}$, $\hat{b}_{u,i,t}$ and $\hat{b}_{u,i,b}$.
Let $v_{e,i}(t) = \begin{bmatrix} v_{e,i,r}(t) & v_{e,i,l}(t) & v_{e,i,t}(t) & v_{e,i,b}(t) \end{bmatrix}^T$ be the vector whose entries are the inputs from neighboring subsystems, the components of $v_{e,i}$ enters the subsystem i with weights respectively $\hat{b}_{i,r}, \hat{b}_{i,l}, \hat{b}_{i,t}, \hat{b}_{i,b}$.
Let $y_{e,i}(t) = \begin{bmatrix} y_{e,i,r}(t) & y_{e,i,l}(t), y_{e,i,t}(t), y_{e,i,b}(t) \end{bmatrix}^T$ be the vector of the outputs, to each outgoing edge it is assigned as usual a unit weight.
Each node of the subsystem have a self-loop, respectively the first and second nodes have self-loops $\hat{\lambda}_{r,i}$ and $\hat{\lambda}_{l,i}$, the third and the forth $\hat{\lambda}_{t,i}$ and $\hat{\lambda}_{b,i}$.
All the nodes of the subsystem are connected to every other node of the same subsystem, let i be the first node and j the second node, a_{ij} is the weight associated to edge (i, j).

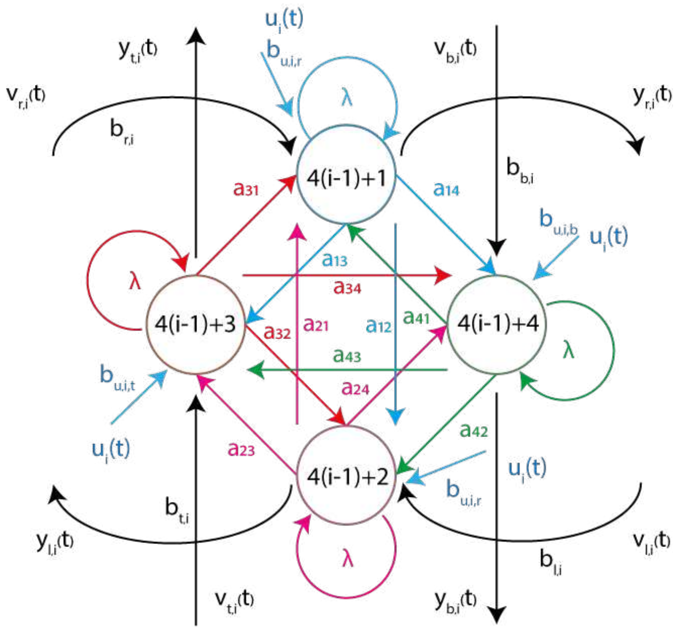

Figure 4.4: Lifted grid subsystem

Thanks to the formulation of equation (2.14) the expanded subsystem can be described by the system of equation:

$$\Sigma_{e,i} : \begin{cases} x_{e,i}(t+1) = A_{e,i}x_{e,i}(t) + B_{e,v,i}v_i(t) + B_{e,u,i}u_i(t) \\ y_{e,i}(t) = C_{e,i}x_{e,i}(t) \end{cases} \tag{4.13}$$

where:

$$\begin{gathered} A_{e,i} = \begin{bmatrix} \hat{\lambda}_{r,i} & \hat{a}_{21,i} & \hat{a}_{31,i} & \hat{a}_{41,i} \\ \hat{a}_{12,i} & \hat{\lambda}_{l,i} & \hat{a}_{32,i} & \hat{a}_{42,i} \\ \hat{a}_{13,i} & \hat{a}_{23,i} & \hat{\lambda}_{t,i} & \hat{a}_{43,i} \\ \hat{a}_{14,i} & \hat{a}_{24,i} & \hat{a}_{34,i} & \hat{\lambda}_{b,i} \end{bmatrix}, \quad B_{e,i} = \begin{bmatrix} \hat{b}_{r,i} & 0 & 0 & 0 \\ 0 & \hat{b}_{l,i} & 0 & 0 \\ 0 & 0 & \hat{b}_{t,i} & 0 \\ 0 & 0 & 0 & \hat{b}_{b,i} \end{bmatrix}, \\ B_{u,i} = \begin{bmatrix} \hat{b}_{u,i,r} & 0 & 0 & 0 \\ 0 & \hat{b}_{u,i,l} & 0 & 0 \\ 0 & 0 & \hat{b}_{u,i,t} & 0 \\ 0 & 0 & 0 & \hat{b}_{u,i,b} \end{bmatrix}, \quad C_{e,i} = I_4. \end{gathered} \tag{4.14}$$

All the subsystems are assumed to be identical by setting:

$$\begin{aligned}&\hat{\lambda}_{r,i} = \hat{\lambda}_{l,i} = \hat{\lambda}_{t,i} = \hat{\lambda}_{b,i} = \hat{\lambda}, \quad \hat{a}_{js,i} = \hat{a}, \quad \hat{b}_{r,i} = \hat{b}_{l,i} = \hat{b}_{t,i} = \hat{b}_{b,i}, \qquad \forall i, j, s \\ &\hat{b}_{u,i,r} = \hat{b}_{u,i,l} = \hat{b}_{u,i,t} = \hat{b}_{u,i,b}.\end{aligned} \tag{4.15}$$

Thanks to this assumption the overall dynamical system can be described by exploiting Kronecker products as:

$$\Sigma_e : \begin{cases} x(t+1) = (I_{nM} \otimes A_{e,i})x(t) + (I_{nM} \otimes B_{e,v,i})v(t) + (B_{e,u,i} \otimes \tilde{B}_{e,u})u(t) \\ y(t) = (I_{nM} \otimes C_{e,i})x(t) \end{cases} \tag{4.16}$$

In order to satisfy the locality constraints we impose:

$$\tilde{B}_{e,u} = \tilde{B}_u, \qquad L_e = L. \tag{4.17}$$

We are hence connecting, as for the simple system, a single input to each row of the grid, the input enters all the nodes in the first subsystem of the row.

The overall state equation for the expanded grid can be written according to equation (2.16) as:

$$\begin{aligned} x_e(t+1) &= \left[I_{nM} \otimes A_{e,i} + (I_{nM} \otimes B_{v,i})L(I_{nM} \otimes C_{e,i})\right]x_e(t) + B_{e,u,i} \otimes \tilde{B}_{e,u}u(t) = \\ &= A_e x_e(t) + B_{e,u}u_e(t). \end{aligned} \tag{4.18}$$

In the section about simulations we will consider the 'symmetric' grid network, which can be obtained from (4.14) by setting:

$$\begin{aligned} &A_{e,i} = \begin{bmatrix} \hat{\lambda} & \hat{a} & \hat{a} & \hat{a} \\ \hat{a} & \hat{\lambda} & \hat{a} & \hat{a} \\ \hat{a} & \hat{a} & \hat{\lambda} & \hat{a} \\ \hat{a} & \hat{a} & \hat{a} & \hat{\lambda} \end{bmatrix}, \quad B_{e,i} = \begin{bmatrix} \hat{b} & 0 & 0 & 0 \\ 0 & \hat{b} & 0 & 0 \\ 0 & 0 & \hat{b} & 0 \\ 0 & 0 & 0 & \hat{b} \end{bmatrix}, \\ &B_{u,i} = \begin{bmatrix} \hat{b}_u & 0 & 0 & 0 \\ 0 & \hat{b}_u & 0 & 0 \\ 0 & 0 & \hat{b}_u & 0 \\ 0 & 0 & 0 & \hat{b}_u \end{bmatrix}, \quad C_{e,i} = I_4. \end{aligned} \tag{4.19}$$

Example 9 (2 × 3 Lifted Grid)
We consider the lift of the grid network with 3 columns and 2 rows described in the previous example. The matrices A_e and $B_{e,u}$ of the system reads:

$$A_e = \begin{bmatrix} A_{e,i} & B_l & 0 & B_t & 0 & 0 \\ B_r & A_{e,i} & B_l & 0 & B_t & 0 \\ 0 & B_r & A_{e,i} & 0 & 0 & B_t \\ B_b & 0 & 0 & A_{e,i} & B_l & 0 \\ 0 & B_b & 0 & B_r & A_{e,i} & B_l \\ 0 & 0 & B_b & 0 & B_r & A_{e,i} \end{bmatrix}, \qquad B_{e,u} = \begin{bmatrix} B_{u,i} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & B_{u,i} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.20}$$

where:

$$B_r = b_t e_{1,4} e_{1,4}^T, \quad B_l = b_l e_{2,4} e_{2,4}^T, \quad B_t = b_t e_{3,4} e_{3,4}^T, \quad B_b = b_b e_{4,3} e_{4,3}^T, \tag{4.21}$$

where $e_{j,4}$ is the j-th canonical vector of dimension 4.

It is possible to appreciate the usefulness of the Kronecker product notation which allow us to connect the structure of the matrix A_e of the lifted system with the matrix A of the simple system. We can see each entry of the matrix A_e as a matrix on his own. If the off diagonal entry A_{jk}, $(j \neq k)$ describes the interconnection between nodes i and j in the simple network, $A_{e,jk}$ describes the same interconnection between subsystems i and j composed of 4 nodes in the lifted network. The diagonal entries of the matrix A_e are the $A_{e,i}$ described in (4.14) (which are assumed to be all identical) and describe the internal dynamics between nodes of the same subsystem.

The local collapsing map describing the induced state by the lifted network is according to equation (2.19):

$$x_{I,i}(t) = \tilde{C}_{e,i} x_{e,i}(t), \tag{4.22}$$

where $\tilde{C}_{e,i} = \mathbb{1}_4^T$.
Instead the state of the underlying network induced by the lift is according to equation (2.18):

$$x_I(t) = I_n \cdot \tilde{C}_{e,i} = I_n \otimes \mathbb{1}_4^T. \tag{4.23}$$

4.2 Simulation and Numerical results

In this section we will perform the energy-related controllability comparison between the lifted grid and the simple grid. Differently form the line network we will limit only to describe the results of some simulations due to the difficulty to find analytic expressions of the metrics for grid networks.
Before discussing the simulations we need to impose the constraints on the allowed dynamics. We have already imposed the same matrices L and $\tilde{B}_u$ for the simple and expanded systems, in this way the locality constraints required by the lift operation are satisfied.
For the computation of Gramians we will solve the discrete-time Lyapunov equations (as for the line network) described by equations (3.43) and (3.44), hence as usual we need to impose Schur stability of the matrices A_e and A.
We will also consider the constraints on the positive-definiteness of $\mathcal{W}$ to impose controllability of the network and the constraints on numerical errors described in section 3.5.
The last constraint we discuss is about the non-amplifying behaviour of the lifted subsystems and we will consider both the requirement on the modulus of eigenvalues for the simple and expanded subsystems and the condition on the asymptotic gains. We compute the eigenvalues of the expanded system in the symmetric configuration, they are given by:

$$\hat{\gamma}_{1,2,3} = \hat{\lambda} - \hat{a}, \qquad \hat{\gamma}_4 = \hat{\lambda} + 3\hat{a}. \tag{4.24}$$

The only eigenvalue of A_i is λ, hence the constrain on the modulus of eigenvalues reads:

$$|\hat{\lambda} - \hat{a}| \leq |\lambda|, \qquad |\hat{\lambda} + 3\hat{a}| \leq |\lambda|. \tag{4.25}$$

The asymptotic gain of the transfer function from the inputs to the collapsed state is given by:

$$W_{e,i}(1) = \begin{bmatrix} \frac{\hat{b}}{1-l-3a} & \frac{\hat{b}}{1-l-3a} & \frac{\hat{b}}{1-l-3a} & \frac{\hat{b}}{1-l-3a} & \frac{\hat{b}}{1-l-3a} & \frac{4\hat{b}_u}{1-l-3a} \end{bmatrix}. \tag{4.26}$$

It is possible to notice by comparing the previous equation with (4.4) that even for the symmetric grid if we choose $\hat{b}_u = \frac{b_u}{4}$, $\hat{b} = b$ and all positive self-loops the constraint on the eigenvalues implies $|W_{e,i}(1)| \leq |W_i(1)|$, and the non-amplifying behaviour is granted also in the transient.

Another configuration we will consider in simulations is $a_{js} = 0 \ \forall (j,s) \notin \Big\{(1,4),(4,1)\Big\}$, we will refer to it as partially connected configuration. In this case the eigenvectors are:

$$\hat{\gamma}_{1,2} = \hat{\lambda}, \qquad \hat{\gamma}_3 = \hat{\lambda} + \hat{a}, \qquad \hat{\gamma}_4 = \hat{\lambda} - \hat{a}. \tag{4.27}$$

The asymptotic gain matrix is:

$$W_{e,i}(1) = \begin{bmatrix} \frac{\hat{b}}{1-\hat{a}-\hat{\lambda}} & \frac{\hat{b}}{1-\hat{\lambda}} & \frac{\hat{b}}{1-\hat{\lambda}} & \frac{\hat{b}}{1-\hat{\lambda}-\hat{a}} & \frac{2b_u(2-2\hat{\lambda}-\hat{a})}{(1-\hat{\lambda})(1-\hat{\lambda}-\hat{a})} \end{bmatrix}. \tag{4.28}$$

Even in this case it is possible to prove that the constraint on the asymptotic gain is granted by the requirement on the modulus of eigenvalues with the same choice of $\hat{b}$ and $\hat{b}_u$ as above.

The simulations will be performed with the same procedure described in section 3.5. The grid can be seen in two dimensions as composed by M rows and n columns. To evaluate the

asymptotic rate for the target networks we need hence to decide how to expand the grid i.e. if we add new rows or columns as we let the number of nodes in the grid grow.
The analysis we are performing is based on the assumption that the number of controlled subsystems remains constant, and since each row of the grid has its own control input we can not change the number of rows, we have to consider grid with an higher number of columns as Mn grows. The dimension of the rows (M) will hence remain constant and we will only chance the number of columns (n).
Before analyzing in detail the results of the simulations it is interesting to point out the reason why we have chosen to add a control input to each row of the network and not only to a single node as for the line. We will motivate our choice with a first simulation.
We consider the simple symmetric grid with the parameter chosen as $\lambda = 0.325, b = 0.16$, we set $M = 2$ and we build all the networks with n taken from 2 to 10. The input is imposed to enter only node 1. For each of these networks we compute the metric $\lambda_{\min}(\mathcal{W})$. The result of the simulation is depicted in figure 4.5.

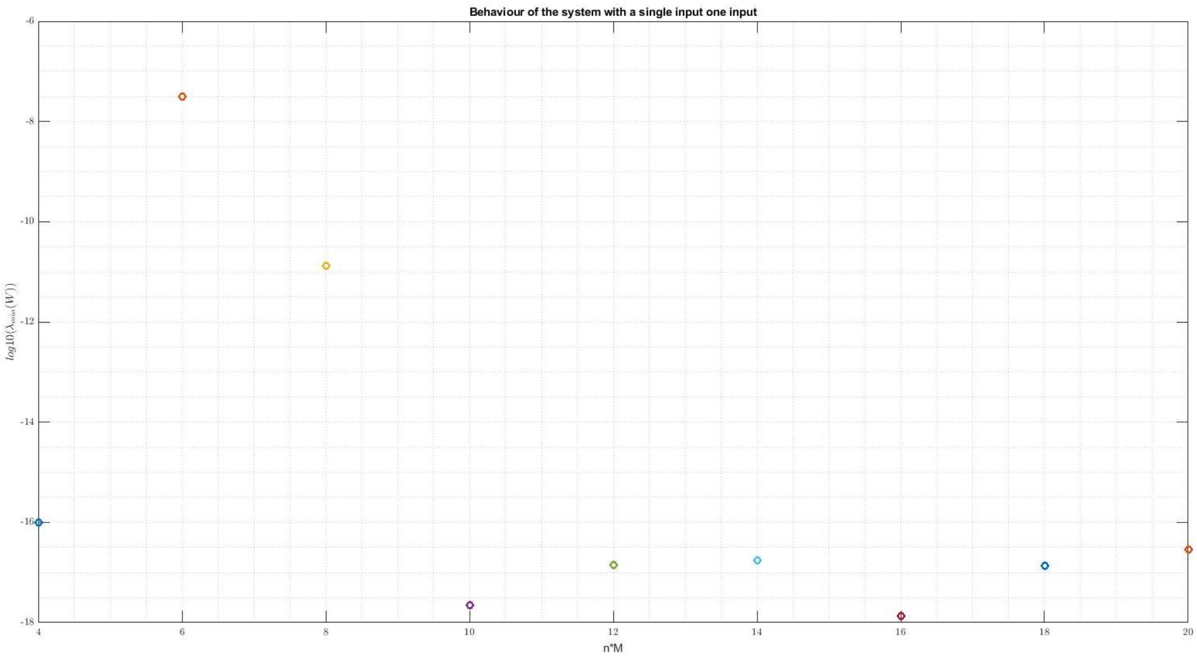

Figure 4.5: Controllability issues with a single input to the grid

From the image above, it is possible to appreciate the behaviour of the metric, it is far from the exponential behaviour (we are taking the logarithm so in the picture we should appreciate a linear decay) we would expect from the theory. Even if structural controllability theory suggest us the system to be structurally controllable, a single input is not sufficient to control the whole grid and hence some controllability problems arises and the metric shows this 'strange' behaviour.
For sake of completeness we recall that the major result of structural controllability theory is that if the system is structurally controllable, it is controllable in the classical sense for **almost** every choice of parameters, therefore there may exist cases in which even if the system is structurally controllable, it is not controllable.
After this short preliminary part we perform some simulations considering the symmetric configuration for simple and expanded grid networks.
As described in the previous section and following the reasoning above, we decide to inject the input to each first node of line in the network. We let n grow, hence we are analyzing the behavior of the network as the number of columns increases and the cardinality of controlled subsystems remains constant, in particular we choose $M = 2$.
As already mentioned, to estimate the asymptotic rate we will follow the same reasoning of section 3.5, in particular we will compute the quantity $e_{s,1}$, $e_{s,2}$ on networks with $n_1 = 2M = 4$

nodes and $n_2 = 4M = 16$ nodes. Such networks have been chosen with a relatively small number of nodes to reduce the possible computational errors.
In the first simulation we highlight a case in which the lifted grid network performs better in terms of energy related controllability than the simple one. We choose the parameters as described in table 4.1, as usual -1:0.005:1 means that the parameter has been chosen in a grid-search fashion.

λ	b	b_u	$\hat{\lambda}$	$\hat{a}$	$\hat{b}$	$\hat{b}_u$
0.325	0.16	1	-1:0.005:1	-1:0.005:1	0.16	0.25

Table 4.1: Choices of the parameters for the first symmetric grid simulation

The result of the simulation is depicted in figure 4.6. For simplicity we plot only the quantity e_2 (which is related to the metric minimum eigenvalue of the Gramian), in fact we know that the behaviour of this metric and the metric trace of the inverse of the Gramian are closely related as Mn grows.
It is possible to notice that the colormap is no more diamond-shaped, this is just because the eigenvalues of the symmetric lifted grid subsystem are different from the ones of the matrix A_e for the line network.
We know the lifted network performs better than the simple one whenever the quantity e_2 assumes positive values. By looking at the color-bar in the figure, the maximum value is greater that 0.1 hence in some cases we have advantages in lifting the dynamics. However by comparing the result with the one achieved for the line, the best parameters for the greed are $\hat{a} = 0$ and $\hat{\lambda} = \lambda$ (we decouple all the line sub-graphs), instead for the line they were $\hat{a} = \lambda$, $\hat{\lambda} = 0$. The two simulations outcomes are not completely in contrast, in fact even in the line case the lifted network performs better than the simple one in the configuration $\hat{\lambda} = \lambda, \hat{a} = 0$, however it is not the best possible choice of parameters.

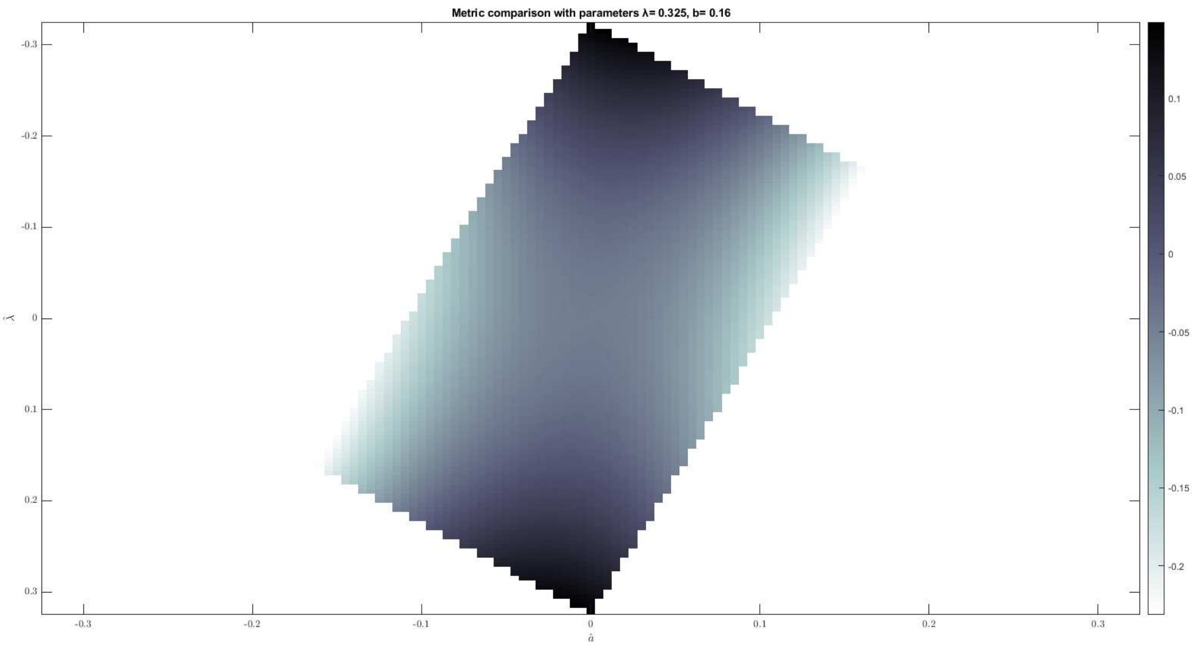

Figure 4.6: Outcome of the first simulation for the symmetric grid

In the second simulation we show there are cases where the asymptotic rate of the lifted network is always greater than the rate of the simple network. We set the parameters as in table 4.2.

λ	b	b_u	$\hat{\lambda}$	$\hat{a}$	$\hat{b}$	$\hat{b}_u$
0.035	0.34	1	-1:0.005:1	-1:0.005:1	0.34	0.25

Table 4.2: Choices of parameters for the second symmetric grid simulation

The results of the simulation are shown in figure 4.7. It is possible to appreciate how the requirement on the modulus of the eigenvalues shrinks the allowed values for $\hat{a}$ and $\hat{\lambda}$ toward the origin. By looking at the colorbar the quantity e_2 is always negative underlying the faster decay of the metric for the lifted network that the simple one. In this lifting the dynamics does not show any advantage in terms of energy related controllability.

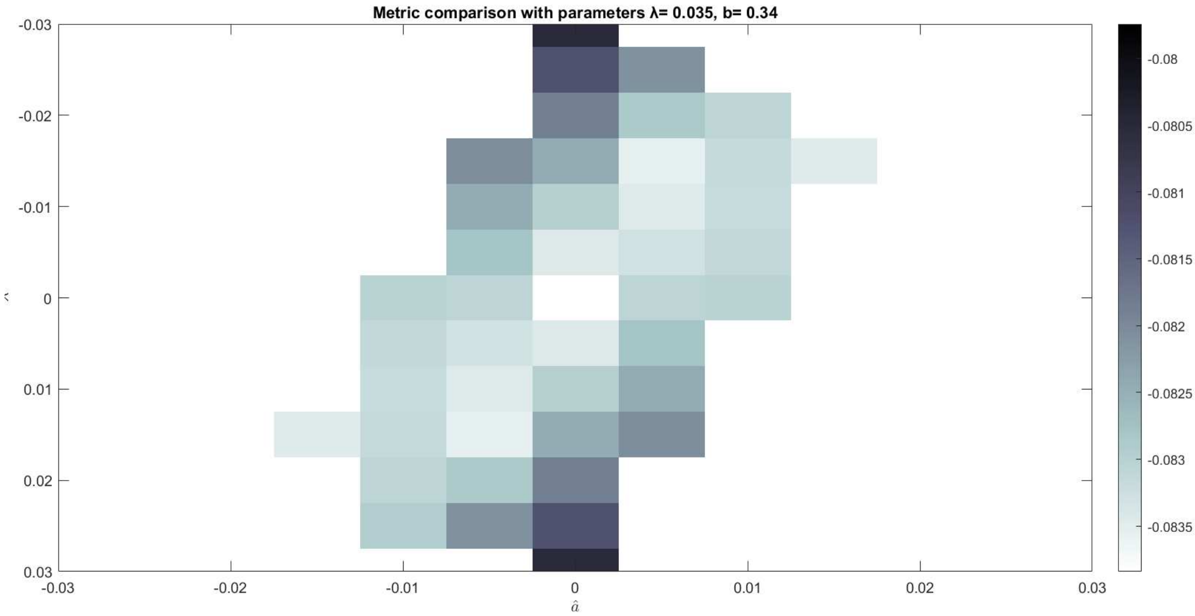

Figure 4.7: Outcome of the second simulation for the symmetric grid

For the last two simulations we consider the partially connected configuration for the lifted grid network. We want to understand what happens if we prevent the interaction between all the nodes in the same subsystem except for a couple of them. To this aim we set $\hat{a}_{js} = 0\ \forall (j,s) \notin \Big\{(1,4),(4,1)\Big\}$, we are allowing the interconnection between the nodes in the grid subsystem which spread the information in the opposite direction with respect to the one from which the input was received. The simulation is performed with the same parameter as before, i.e., as in table 4.1, with the difference that we change only the entries $\hat{a}_{1,4}, \hat{a}_{4,1}$ of the matrix A_e according to the parameter $\hat{a}$. The simulation outcome is shown in figure 4.8. It is possible to notice again the diamond-shaped color-map due to the eigenvalues of A_d. Even in this case the best result is achieved by decoupling all the nodes in the same subsystem.

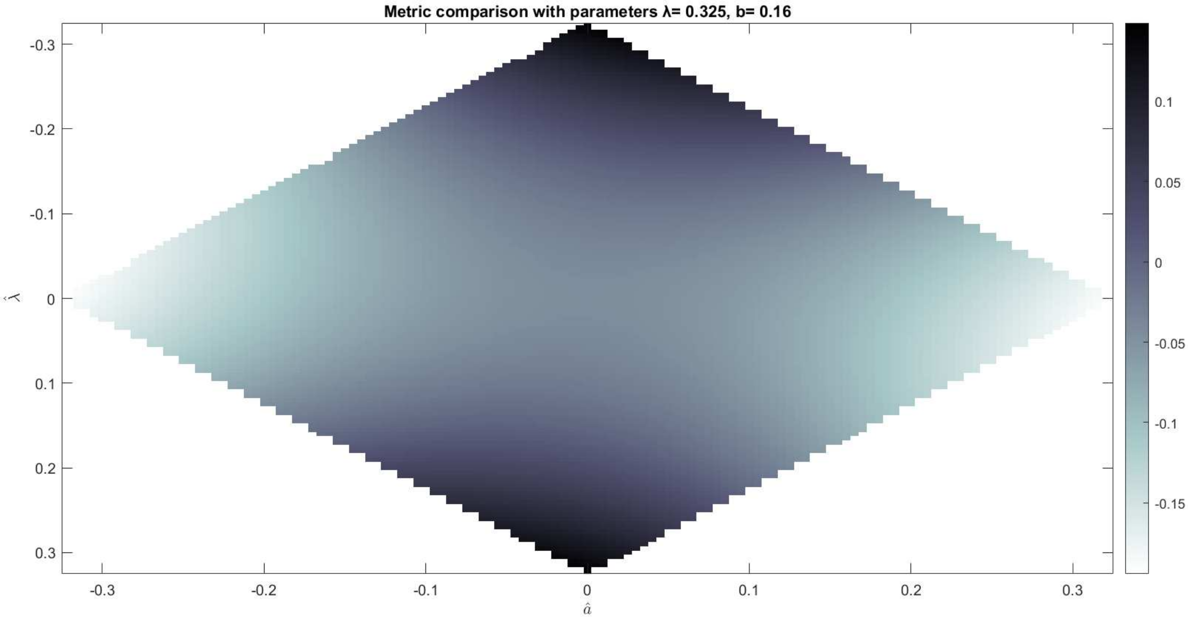

Figure 4.8: Results of the first simulation for the partially connected grid

We end this section by considering the same configuration for the lifted grid network of before and we show that even in this case there are parameters choices which does not lead to any advantage in using the lifted configuration with respect to the simple one. We set the parameters according to table 4.2, with the exception that only $\hat{a}_{1,4}, \hat{a}_{4,1}$ of the matrix A_e are set according to the parameter $\hat{a}$, all the other a_{js} remains 0 for all j, s. The result of the simulation is shown in figure 4.9, even if the best results are achieved again for the lifted configuration with $\hat{\lambda} = \lambda$ and $\hat{a} = 0$, the quantity e_2 is always negative highlighting the fact that the metric minimum eigenvalue of the Gramian for the simple network decays more slowly with respect to the lifted counterpart.

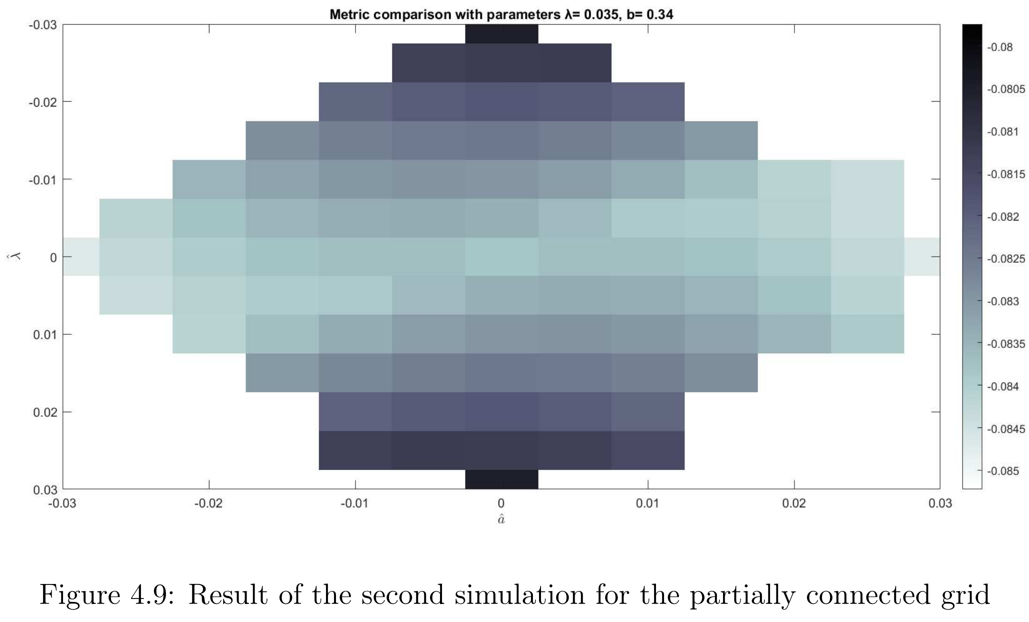

Figure 4.9: Result of the second simulation for the partially connected grid

It is worth noting that the outcomes of the simulations in the partially connected configuration are quite similar (except for the shape of the allowed parameters region imposed by the eigenvalues of A_e) to the ones obtained with the symmetric configuration.

Conclusions

In this thesis, we have discussed the role of structure and memory in reducing the effort necessary to control a networked dynamical system as the number of control inputs remains constant and the size of the network increases. Each node in the network was embedded with a local memory on the source direction of incoming control inputs. Furthermore, nodes were endowed with computational power to process inputs and efficiently propagate them through the network. The key theoretical tool that allowed us to enrich the internal dynamics of the nodes and add new control capabilities was graph lifting. In essence, graph lifting consists in expanding (adding components) to the original node state by generating a novel graph composed of the same number of subsystems (sub-graphs) as the number of nodes in the original non-lifted network. Each lifted subsystem can hence be formed by more than one node, and the joint state of all the nodes in the same lifted subsystem represents the expanded node state.
In the thesis we have described in detail the lifting procedure and how it can be applied to linear dynamical systems on networks. In particular we focused on homogeneous lifting, where each sub-graph state has the same dimension and is subjected to identical dynamical equations. Some key constraints on the allowed lifted dynamics were introduced and discussed, in order to prevent the expanded dynamics from amplifying input signals more than the original one. These constraints play a crucial role in the interpretation of the achieved results.
After a general description of the framework the attention was dedicated to some specific topologies, the line and the grid, which are used as case studies. For these topologies we discussed how to "design" the lifted network to reproduce memory effects.
In the line case, once treated how the simple and lifted lines can be modeled, we computed analytical expressions of the control energy metrics and of the asymptotic rates (rate of growth of the energy metrics as the number of nodes in the network increases) for both the networks. Furthermore, an inequality has been derived which allows to prove that for a specific selection of free parameters the lifted network outperforms the simple one. The considered choice of parameters satisfies the most conservative constraints for the expanded system, and alternative parameter choices are possible, also yielding good control energy performances for the lifted line. However, it was outlined through some examples how there exist sets of parameters of the initial non-lifted network that do not allow for any improvement in terms of control energy by employing the proposed lifted dynamics. The analytical results were followed by grid-search simulations in order to evaluate the best parameters for the lifted network beyond the one leading to the inequality. As expected, these simulations confirmed how the set of parameters considered in the inequality does not yield to the best performances of the lifted line.
The focus subsequently shifted to describe how to embed memory in the grid network. Given the complexity of analytical computations for control energy metrics, advantages were evaluated through simulations. The conducted simulations highlighted again the convenience of employing the lifted dynamics. Nevertheless, it is worth noting that, as for the line, certain configurations for the original non-lifted grid do not allow to improve control energy performances by adding memory.
Summing up, with this work we proposed a first description of the potential of memory effects for the control of networked dynamics. We managed to prove how these effects can be effectively designed and used to reduce the effort necessary to control a network system. However, we pointed out that it is not always possible (at least with the imposed constraints) to employ memory to improve control energy performances.

A number of issues still remain open for further investigation. For example, numerical simulations for the grid and the line seem to lead to different conclusions on the best choice of parameters for the lifted dynamics. This may depend on the structure of the network. However, by suitably loosening the constraints, the results of the simulations are more similar. As we derived them, the imposed constraints on the dynamics are meaningful only if we consider as inputs to the system step signals. Further investigation on the role and the proper form of the constraints can hence be considered to understand if there exist better conditions capturing the required non-amplifying behaviour. Another open problem is related to the existence of solutions that do not lead to any advantage in employing memory effects, the sets of parameters showing this behaviour were not fully characterized. Finally, analytical expressions of asymptotic rates could be derived for alternative parameter choices (beyond the considered one), in particular for the ones resulting in the lowest asymptotic rates. This would provide a more precise quantification of the benefits derived from employing the proposed strategy.

Appendix A

Useful tools

A.1 Kronecker Product

The Kronecker product is a useful tool we will exploit for the modeling of network of interconnected homogeneous systems. In this appendix we provide some basic definitions which will help the reader to better understand the content of the thesis work.

Definition 15 (Kronecker Product) *Consider two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{q\times r}$, the Kronecker product of A and B is the $nq \times mr$ matrix $A \otimes B$ given by:*

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \ddots & a_{nm}B \end{bmatrix}. \tag{A.1}$$

If we consider two general vectors $v, w \in \mathbb{R}^n$, we have that:

$$v \otimes w = \begin{bmatrix} v_1 w \\ \vdots \\ v_n w \end{bmatrix} \in \mathbb{R}^{n^2}. \tag{A.2}$$

The Kronecker product enjoys several properties, we list only some of them below:

- bilinearity property: $(\alpha A + \beta B) \otimes (\gamma C + \delta D) = \alpha\gamma A \otimes C + \alpha\delta A \otimes D + \beta\gamma B \otimes C + \beta\gamma B \otimes D$
- associativity property: $(A \otimes B) \otimes C = A \otimes (B \otimes C)$
- the transpose property: $(A \otimes B)^T = A^T \otimes B^T$
- the mixed product property: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

Where A, B, C, D are all matrices with appropriate compatible dimensions.
Some useful consequences of the mixed product property concerns eigenvalues of the product, let $Av = \lambda v$ and $Bw = \mu$, then:

- the eigen property reads: $(A \otimes B)(v \otimes w) = (Av) \otimes (Bw) = (\lambda v) \otimes (\mu w) = \lambda\mu(v \otimes w)$
- the spectrum property reads: $\sigma(A \otimes B) = \{\lambda\mu | \lambda \in \sigma(A), \mu \in \sigma(B)\}$

The last property we will consider is the inverse property. For square matrices A and B, $A \otimes B$ is invertible if and only if both A and B are inevitable. In this case the inverse property hold:

- inverse property $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$

To better develop the introduced concepts we provide a simple example of use of Kronecker product.

Example 10 (Kronecker Product)
Consider the two matrices:

$$A = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix}. \tag{A.3}$$

The Kronecker product is given by:

$$A \otimes B = \begin{bmatrix} 3B \\ 2B \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 6 & 21 \\ 2 & 0 \\ 4 & 14 \end{bmatrix}. \tag{A.4}$$

A.2 Discrete-Time Lyapunov equations

In the thesis we will often perform computations of the infinite horizon controllability and output controllability Gramians. They require the solution of the discrete time Lyapunov equation. We report here the definition and the conditions which leads to a unique solution of the equation.
The discrete-time Lyapunov equation is

$$AXA^{\dagger} - X = -Q \tag{A.5}$$

where Q is hermitian.

1. There is a unique solution iff no eigenvalue of A is the reciprocal of an eigenvalue of $A^{\dagger}$. If this condition is satisfied, the unique X is Hermitian.
2. If A is convergent then X is unique and Hermitian and $X = \sum_{k=0}^{\infty}(A^k Q (A^{\dagger})^k)$.
3. If A is convergent and Q is positive definite (or semi-definite) then X is unique, Hermitian and positive definite (or semi-definite).

We recall that a matrix A is convergent if A^k tends to 0 as k tends to infinity. In order to establish if a matrix is convergent it is possible to check its eigenvalues, all the eigenvalues needs to be inside the unit circle, i.e. let $\sigma(A)$ be the spectrum of A, A is convergent iff $\lambda \in \sigma(A) \implies |\lambda| < 1$.
A matrix whose eigenvalues are all inside the unit circle is said to be Schur stable.
Given a system described by the matrices (A, B) the infinite-horizon controllability Gramian is given by the equation:

$$\mathcal{W} = \sum_{k=1}^{n} A^k B B^T (A^T)^k. \tag{A.6}$$

In particular we will always consider real matrices, hence $A^T = A^{\dagger}, B^T = B^{\dagger}$.
The Gramian can hence be found using Lyapunov equations by exploiting point 2 above, in particular it is necessary to set:

$$X = \mathcal{W}, \quad Q = BB^T \tag{A.7}$$

The matrix Q defined in this way will always be positive semi-definite, moreover if we impose Schur stability of the matrix A (as we will always do in the thesis) the solution of the Lyapunov equations and hence the Gramian will be positive semidefinite and unique.

A.3 Asymptotic gain of a discrete time transfer function

Consider a system with rational transfer function

$$W(z) = \frac{a(z)}{b(z)} \tag{A.8}$$

We assume this function to be BIBO stable and proper.
We consider the discrete step signal:

$$u(t) = \delta_{-1}(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{A.9}$$

The corresponding $\mathcal{Z}$-Transform is a geometric series of ratio z^{-1}, therefore we have:

$$U(z) = \mathcal{Z}[\delta_{-1}(t)] = \sum_{k=0}^{+\infty} z^{-k} = \frac{1}{1 - z^{-1}} = \frac{z}{z-1}. \tag{A.10}$$

If we apply the step signal to the system the forced response to this input is:

$$Y(z) = W(z)\frac{z}{z-1}. \tag{A.11}$$

Let $y(t) = \mathcal{Z}^{-1}[Y(z)]$, we can now compute the asymptotic value of the signal which is given according to the final value theorem for the $\mathcal{Z}$-transform.
Theorem 4 (Final Value theorem) *Consider a signal f(t), and let $F(z) = \mathcal{Z}[f(t)]$ be its $\mathcal{Z}$-transform. if $\lim_{t\to\infty} f(t)$ exist and is finite, then:*

$$\lim_{t\to\infty} f(t) = \lim_{z\to 1}(1 - z^{-1})F(z). \tag{A.12}$$

By exploiting the final value theorem we can compute the asymptotic value for the signal $y(t)$ which is also called **asymptotic gain**:

$$\lim_{t\to\infty} y(t) = \lim_{z\to 1}(1 - z^{-1})Y(z) = \lim_{z\to 1}(1 - z^{-1})\frac{z}{z-1}W(z) = W(1) \tag{A.13}$$

Hence given a transfer function $W(z)$ the asymptotic gain is simply given by $W(1)$, it represents the asymptotic value of the forced response to the step signal. If $W(1) > 1$ it means that the system amplifies step input signals, if $W(1) < 1$ the signal is attenuated.

Appendix B

Matlab Code

B.1 Simulations for simple and expanded line networks

The following Matlab code implements the procedure for the estimation of the asymptotic rates described in section 3.5. Given a certain set of parameters for the simple line the code generates the color-maps used to evaluate the best parameters choices for the associated lifted line.
In the script the parameters λ for the simple line will be denoted as l; the parameters $\hat{\lambda}$, $\hat{a}$, $\hat{b}$ for the expanded line as lh, ah and bh.

```
clc
clear all
close all
%parameters of the simple line for which tehere exists
%advantageous configuarations of the associated lift
l= 0.2;%self loop weights
b= 0.39;% edge weights
bu=1 %input weight
%parameters of the lifted line for which does not exist
% advantageous configurations of the associated lift
%l= 0.2;%self loop weights
%b= 0.05;% edge weights
%bu=1 %input weight
%% parameteres for the expanded network
bh=b; %edge weights
buh=0.5 %interconnection weights
%% selection of the contraints to be applied
%set contrain = 1 to activate the contrain
constr_eigenvalues=1;
constr_as_gain=1;
%The vector energy[1,2] describes the control enegy using
%two different metrics
%Let gmin=min(eig(W)); tw=trace(inv(W))
%We compute the control energy for two networks
%one with 12 nodes (gmin12 and tw12)
%and  one with 16 nodes (gmin16 and tw16)
%We do the same for the associated expanded networks
%(gmin12_e and tw12_e, gmin16_e and tw16_e).
%Let e1=log10(tw16)-log10(tw12)
%Let e2=log10(gmin16)-log10(gmin12)
%Let e1_e=log10(tw16_e)-log10(tw12_e)
%Let e2_e=log10(gmin16_e)-log10(gmin12_e)
%the two vectors energy1 and energy2 are structured as follows
```

```matlab
%-----------------------------------------------
%                 Energy1-Simple network
%-----------------------------------------------
% |-l-|-b-|-tw12-|-gmin12-|-tw16-|-gmin16-|-tw16-|
%------------------------------------------------
%                 Energy2-Expanded Network
%------------------------------------------------
%|-lh-|-ah-|-tw12_e-|-gmin12_e-|-tw16_e-|-gmin16_e-|-tw16_e-|
energy1=[l b];
%% computation of the control energy for the simple networks with 12 and 16 nodes
for n=[12,16]
        % we build the simple line network
        A = l*diag(ones(n,1)) + b*diag(ones(n-1,1),1) + b*diag(ones(n-1,1),-1);
        B = zeros(n,1);
        B(1) = bu;
        if(max(abs(eig(A)))>1) %we check the stability of matrix A
            ME = MException('My:unstable','A is not stable');
            throw(ME)
        end
%computation of the gramian by solving Lyapunov equations
        W = dlyap(A,B*B');
%We check the positive definitness of the gramian wich combined
%with the stability of A implies  the network to be controllable
        if(find(eig(W))<=0)
                ME = MException('My:posdef','W is not positive definite');
                throw(ME)
        end
%we add the entries to the vector energy1
        energy1 = [energy1 trace(inv(W)) min(eig(W))];
 end
%% computation of the control energy for the expanded network
energy2=[];
%-------------------------------------------------------------------
% variables necessary to generate the colormap
itl=1;
ita=1;
s=length(-1:0.01:1);
e1c=-1e10*ones(s,s);
e2c=1e10*ones(s,s);
max_e1=-1e10;
min_e2=1e10;
% ------------------------------------------------------------------
% simple subsystem matrices for the comparison of asymptotic gains
Ane=l;
Bne=[b b 1];
% grid search on the parameters for the expanded network
for lh = -1:0.01:1
    for ah=-1:0.01:1
        stop=false;
        energy_t=[lh ah];
```

```
        for n=[12,16]
%matrices of the expanded subsystem for the
%asymptotic gain computation
            Aei=[lh ah; ah lh];
            Bevi=[bh 0 buh;
                0 bh buh];
            Ctei=ones(1,2);
%we check the contraints on the modulus of eigenvalues
            if(constr_eigenvalues==1)
                 ee=eig(Aei);
                 for i=1:length(eig(Aei))
                      if(abs(ee(i))>abs(l))
                          stop=true;
                      end
                  end
             end
%asymptotic gain computation for the expanded
%and simple systems
            We=Ctei*inv(Aei-eye(2))*Bevi;
            Wne=inv(Ane-1)*Bne;
            %asymptotic gain comparison
             for j=1:3
                if(abs(We(1,j))>abs(Wne(1,j)))
                     stop=true;
                end
             end
% We build the matrix Ae of the overall expanded system
            Ae = [lh*diag(ones(n,1))+bh*diag(ones(n-1,1),1) ah*eye(
               n); ah*eye(n) lh*diag(ones(n,1))+bh*diag(ones(n-1,1)
               ,-1)];
%we check the stability of matrix Ae
            if(max(abs(eig(Ae)))>=1)
                    stop=true;
            else
            Be = zeros(2*n,1);
            Be(1) = buh;
            Be(n+1) = buh;
            Ct = [eye(n) eye(n)];
%output controllability gramian computation
            Wdt = Ct*dlyap(Ae,Be*Be')*Ct';
%we check the positive definitness of the gramian
            if(find(eig(Wdt))<=0)
                ME = MException('My:posdef','W is not positive
                   definite');
                throw(ME)
            end
%we add the entries to a temporary vector
            energy_t = [energy_t trace(inv(Wdt)) min(eig(Wdt))];
            end
        end
%if all the contraints are satisfied we add the entries in
%energy_t to the vector energy_2
            if(stop==false)
```

```
                    energy2=[energy2;energy_t];
%we create the matrices of the quantities e1 and e2
% to generate the colormap
                    ee1=log10(energy_t(:,5))-log10(energy_t(:,3));
                    es1=log10(energy1(5))-log10(energy1(3));
                    e1=ee1-es1;
                    e1c(itl,ita)=e1;
                    if(imag(e1c(itl,ita))==0)
                        e1c(itl,ita)=e1;
                    end
                    ee2=log10(energy_t(:,6))-log10(energy_t(:,4));
                    es2=log10(energy1(6))-log10(energy1(4));
                    e2=ee2-es2;
                    if(imag(e2c(itl,ita))==0)
                         e2c(itl,ita)=e2;
                    end
               end
          ita=ita+1;
     end
     ita=1;
     itl=itl+1;
end
%%We plot the log - control energy comparison with both the metrics
     in a colormap
e1c(find(e1c==-1e10))=max(e1(:,:));
e2c(find(e2c==1e10))=min(e2(:,:));
max_l=find(max(real(energy2(:,1))));
min_l=find(min(real(energy2(:,1))));
l_limit=max(abs(energy2(max_l,1)),abs(energy2(min_l,1)))
max_a=find(max(real(energy2(:,2))));
min_a=find(min(real(energy2(:,2))));
a_limit=max(abs(energy2(max_a,1)),abs(energy2(min_a,1)))
%we generate the colormap for the metric trace of the gramian
figure
title(sprintf('Metric comparison with parametrs $\lambda$=%d, b=%d'
    ,l,b))
map1=subplot(2,1,1);
im=image(-1:0.01:1,-1:0.01:1,e1c);
xlim([-a_limit, a_limit])
ylim([-l_limit, l_limit])
colorbar
c=colormap("bone");
colormap(map1,c);
im.CDataMapping = 'scaled';
title('Metric $Trace(W^{-1})$','Interpreter','latex')
xlabel('$$\hat{a}$$','Interpreter','latex')
ylabel('$$\hat{\lambda}$$','Interpreter','latex')
%we generate the colormap for the metric minimum eigenvalue of the
    gramian
map2=subplot(2,1,2);
im=image(-1:0.01:1,-1:0.01:1,e2c);
colorbar
colormap(map2,flipud(c))
```

```
im.CDataMapping = 'scaled';
xlabel('$$\hat{a}$$','Interpreter','latex')
ylabel('$$\hat{\lambda}$$','Interpreter','latex')
xlim([-a_limit, a_limit])
ylim([-l_limit, l_limit])
title('Metric $\lambda_{min}(W)$','Interpreter','latex')
sgtitle(sprintf('Metric comparison with parameters  = %.03f, b=
    %.02f',l,b))
```

B.2 Simulations for the grid network

In this section we report the code used for the estimation of the asymptotic rates for the simple and lifted grid networks in the fully-connected symmetric configuration. The simulations for the partially connected configuration can be obtained by adapting the code. We exploit Kronecker products to compute the matrices of the overall systems starting from the matrices of the simple and expanded subsystems. In order to estimate the asymptotic rates we exploit the procedure outlined in sections 3.5 and 4.2. The last rows of the script plot the color-map for the evaluation of the best parameters for the lifted grid network.

```
clc
clear all
close all
%parameters of the simple grid for which tehere exists
%advantageous configuarations of the associated lifted grid
l=0.325; %parameter lambda, selfloop weight
b=0.16; %interconnection weight
bu=1; %weight associated to the input from external enviroment
%parameters of the simple grid for which does not exists
%advantageous configuarations of the associated lifted grid
% l=0.035;
% b=0.34;
%paramteres of the lifted grid
bh=b %parameter hat b, interconnection weight
beu=1/4 %%weight associated to the input from external enviroment
% dimesions of the two grid for the rate estimation
n1=2; m1=2;
n2=4; m2=2;
%estimation of the asymptotic rate for the simple grid
rate1=compute_sg_rate(n1,m1,n2,m2,l,b,bu);
%vector containing the difference between asymptotic
%rates i.e the quantity e2
energy_e=[];
etl=1;eta=1;
%% grid search for the parameters of the lifted grid
for lh=-1:0.005:1
    eta=1;
    for ah=-1:0.005:1
        try
%the expanded grid is built
            [A1 B1 C1]=build_grid_ex(n1,m1,lh,bh,ah,beu);
            [A2 B2 C2]=build_grid_ex(n2,m2,lh,bh,ah,beu);
%computation of the gramians
            W1=out_gramian(A1, B1, C1);
            W2=out_gramian(A2, B2, C2);
%matrices describing the expanded subsystem
```

```
            Aei=lh*eye(4)+ah*ones(4,4)-ah*eye(4);
            Bei=[bh*eye(4) beu*ones(4,1)];
%matrices describing the simple subsystem
            Ai=l;
            Bi=[b*ones(1,4) 1];
%check of gramian positive definitness
            if(min(eig(W1))<=0 && min(eig(W2))<=0)
               ME=MException('W:unct','unct W');
               throw(ME)
            end
%check of the contraint on eigenvalues
            ee=eig(Aei);
            for i=1:length(ee)
                if(abs(ee(i))>abs(l))
                    ME=MException('A:eigenvals','constaint on eigenvalues not satisfied');
                    throw(ME)
                end
            end
%check of the contrain on asymptotic gain
            We=ones(1,4)*inv(Aei-eye(4))*Bei;
            Wne=inv(Ai-1)*Bi;
             for i=1
                 for j=1:5
                    if(abs(We(i,j))>abs(Wne(i,j)))
                         ME=MException('We:asgain','constraint on asymptotic value not satisfied');
                         throw(ME)
                    end
                 end
             end
%computation of the as. rate for the expanded grid
            rate=log10(min(eig(W2)))-log10(min(eig(W1)));
%we check for numerical errors
            if(imag(rate)==0)
            else
                ME=MException('We:imag','imag');
                throw(ME)
            end
%we add the computed rate to the vector energy_e
            energy_e=[energy_e; lh ah rate-rate1 etl eta];
        catch ME
        end
        eta=eta+1;
    end
    etl=etl+1;
end
%% colormap genertation
s=length(-1:0.005:1);
e=1e10*ones(s,s);
max_e=1e10;

max_l=find(max(real(energy_e(:,1))));
```

```
min_l=find(min(real(energy_e(:,1))));
l_limit=max(abs(energy_e(max_l,1)),abs(energy_e(min_l,1)))

max_a=find(max(real(energy_e(:,1))));
min_a=find(min(real( energy_e(:,1))));

a_limit=max(abs(energy_e(max_a,1)),abs(energy_e(min_a,1)))

for i=1:size(energy_e)
    eta=energy_e(i,5);
    etl=energy_e(i,4);
    e(etl,eta)=real(energy_e(i,3));
end

max_e=max(energy_e(:,3));
min_e=min(energy_e(:,3));
e(find(e==1e10))=min_e;
figure
 title('Metric trace(W^{-1})')
 im=image(-1:0.005:1,-1:0.005:1,e);
 xlim([-a_limit, a_limit])
 ylim([-l_limit, l_limit])
colorbar
 c=colormap("bone");
 colormap(flipud(c))
 im.CDataMapping = 'scaled';
 clim([min_e, max_e])
 xlabel('$\hat{a}$','Interpreter','latex')
 ylabel('$\hat{\lambda}$','Interpreter','latex')
 title(sprintf('Metric comparison with parameters   = %.03f, b=
    %.02f',l,b))
function [A,B,C]=build_grid_ex(n,m,lh,bh,a,beu)
%function to build the expanded grid given the parameters
    Aei=lh*eye(4)+a*ones(4,4)-a*eye(4);
    Bei=bh*eye(4);
    Cei=eye(4);
    e=@(k,n) [zeros(k-1,1);1;zeros(n-k,1)];
    L=(kron(kron(eye(m),diag(ones(n-1,1),1)),e(2,4)*e(2,4)')+kron(
       kron(eye(m),diag(ones(n-1,1),-1)),e(1,4)*e(1,4)')+kron(kron(
       diag(ones(m-1,1),-1),eye(n)),e(4,4)*e(4,4)')+kron(kron(diag(
       ones(m-1,1),1),eye(n)),e(3,4)*e(3,4)'));
    A=kron(eye(n*m),Aei)+kron(eye(n*m),Bei)*L*kron(eye(n*m),Cei);
    B=beu*kron(kron(eye(m),[1;zeros(n-1,1)]),ones(4,1));
    C=kron(eye(n*m),ones(4,1)');
end
function rate= compute_sg_rate(n1,m1,n2,m2,l,b,bu)
%function which computes the asymptotic rate for the simple line
    [A1 B1]=build_grid(n1,m1,l,b,bu,1);
    W1=dlyap(A1,B1*B1');
    e11=log10(min(eig(W1)));
    [A2 B2]=build_grid(n2,m2,l,b,bu,1);
    W2=dlyap(A2,B2*B2');
    e21=log10(min(eig(W2)));
```

```
    rate=e21-e11
end
function [A,B]=build_grid(n,m,l,b,bu,f)
%function that builds the simple grid given the parameters
    Ai=l;
    Bi=b*ones(1,4);
    Ci=ones(4,1);
    e=@(k,n) [zeros(k-1,1);1;zeros(n-k,1)];
    L=(kron(kron(eye(m),diag(ones(n-1,1),1)),e(2,4)*e(2,4)')+kron(
       kron(eye(m),diag(ones(n-1,1),-1)),e(1,4)*e(1,4)')+kron(kron(
       diag(ones(m-1,1),-1),eye(n)),e(4,4)*e(4,4)')+kron(kron(diag(
       ones(m-1,1),1),eye(n)),e(3,4)*e(3,4)'));
    A=kron(eye(n*m),Ai)+kron(eye(n*m),Bi)*L*kron(eye(n*m),Ci);
    B=kron(eye(m),[bu;zeros(n-1,1)]);
    %if the flag f=1 we check for the stability of A
    if(f==1)
         if(round(max(abs(eig(A))),15)>=1)
                ME=MException('A:unst','unstable A');
                throw(ME)
         end
    end
end
function [W] = out_(A,B,C)
%function that computes the output controllability gramian
    if(max(abs(eig(A)))>=1)%we chek for the stability of Ae
        ME=MException('Ae:unst','unstable Ae');
        throw(ME)
    end
    W=C*dlyap(A,B*B')*C';
end
```

Bibliography

[1] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, pp. 40–52, 06 2014.

[2] Y.-Y. Liu and A.-L. Barabasi, "Control principles of complex networks," *Rev. Mod. Phys.*, vol. 88, 08 2015.

[3] S. Apers, F. Ticozzi, and A. Sarlette, "Lifting markov chains to mix faster: Limits and opportunities," 05 2017.

[4] P. Diaconis, S. P. Holmes, and R. M. Neal, "Analysis of a nonreversible markov chain sampler," *Annals of Applied Probability*, vol. 10, pp. 726–752, 2000.

[5] F. Bullo, *Lectures on Network Systems.* Kindle Direct Publishing, 1.6 ed., 2022.

[6] S. Noschese, L. Pasquini, and L. Reichel, "Tridiagonal toeplitz matrices: Properties and novel applications," *Numerical Linear Algebra with Applications*, vol. 20, 03 2013.

[7] G. Baggio and S. Zampieri, "Worst-case control energy of single-input network systems," *IFAC-PapersOnLine*, vol. 55, no. 13, pp. 55–60, 2022.

[8] G. Baggio and S. Zampieri, "Controllability of large-scale networks:the control energy exponents," *IEEE Transactions on Control of Network Systems (to appear)*, 2023.

[9] K. M. Schmidt and A. Zhigljavsky, "A characterization of the arcsine distribution," *Statistics & Probability Letters*, vol. 79, pp. 2451–2455, 12 2009.

[10] E. Fornasini, *Appunti di teoria dei sistemi / E. Fornasini.* Padova: Progetto, 2015.

[11] H. M. Taylor, *An introduction to stochastic modeling.* San Diego, California ;: Academic Press, Inc., revised edition. ed., 1994 - 1994.

[12] M. Brookes, "The matrix reference manual," 2020.

[13] M. Pavon, *Lectures on Finite Probabilistic Models.* 2008.

[14] M. Fiedler, "Notes on hilbert and cauchy matrices," *Linear Algebra and its Applications*, vol. 432, no. 1, pp. 351–356, 2010.

[15] G. B. M. B. A. F. F. Ticozzi and S. Zampieri, *Lecture Notes on Digital Control.* 2021.

List of Figures