UNIVERSITY OF GRONINGEN

BACHELOR INTEGRATION PROJECT

Topology identification of complex networks of linear systems

Author: Rowan Timmermans (s2740192) Supervisors: Prof. Dr. Claudio de Persis Dr. Ir. Gerald H. Jonker Henk J. van Waarde (PhD)

June 17, 2019



Abstract

Topology identification is an important tool to facilitate analyzing, predicting and controlling network behaviour. In the following thesis, we study topology identification for networks with general time-invariant linear node dynamics, and we propose an identification approach based on subspace identification methods.

The approach is derived in two steps. First, the network dynamics are determined based on input/output (I/O) data, using a subspace identification algorithm. Then, under suitable identifiability conditions, the network topology can be obtained by solving a system of linear equations.

The resulting method has been coded in MatLab and validated by means of an example network. We conclude that a network's topology can be identified based on measured input/output data using a subspace identification method, assuming that the local node dynamics are known.

Contents

1	ntroduction	3	
2	Problem analysis 9.1 Problem context 9.2 Problem owners and stakeholders 9.3 System and scope 2.3.1 System description 2.3.2 Scope 2.4 Research goal 2.5 Tools and methods 2.5.1 Literature research 2.5.2 MatLab 2.5.3 Cycle choice	3 3 4 4 4 6 6 6 6 7 7	
3	Results 3.1 Compact formulation of network dynamics 3.2 Subspace identification 3.2.1 Rewriting the model 3.2.2 Matrix identification methods 3.2.3 State vector sequence approximation 3.2.4 Similarity transformation 3.3 Implementation in topology identification	7 8 9 10 11 12 12	
4	Alidation	14	
5	Conclusion	16	
6	Discussion and further research	16	
Re	References		
A	Appendices		
Α	Matlab codesA.1 MatLab code of topology identificationA.2 MatLab code of subspace identification algorithm	19 19 22	

1 Introduction

Nowadays, complex networks exist in every corner of the world, from communication networks to social networks, from cellular networks to metabolic networks, from Internet to World Wide Web [1]. Networks have two main aspects:

- Nodes: these represent subsystems that each have their own dynamics.
- Edges: these represent communication or information exchange between nodes.

Analysis and control of the behaviours of complex networks consisting of a large number of dynamical nodes have attracted wide attention in the past decade. In particular, special attention has been given to the control and synchronization of large scale complex dynamical networks with certain types of topology [2].

However, in the real world, the exact topology of a complex dynamical network is often unknown or uncertain [3]. The procedure of finding a network's topology is called topology identification. Much research has already been conducted to investigate new approaches to this topic. However, many of these methods exist under the assumption of relatively simple node dynamics. For example, M. Nabi-Abdolyousefi and M. Mesbahi developed a network identification technique assuming single integrator node dynamics [4]. Then, in [5], this technique was extended for networks of systems that are not necessarily single integrators, however are still assumed to be SISO and identical.

We refer to a network with identical subsystems as a *homogeneous* network. The more general setting in which the node dynamics are not necessarily the same is referred to as a *heterogeneous* network. In the following thesis, we study topology identification of heterogeneous networks with general linear node dynamics; the systems are not necessarily SISO.

2 Problem analysis

2.1 Problem context

Consider a network of neurons in the brain. While each neuron has its own internal dynamics, the dynamics of the network as a whole are what makes the brain function as it is [6]. Additionally, large power grids rely on stability and controllability of the interconnections within the grid to reduce the probability of power outages [7].

These examples illustrate the importance of topology identification; understanding and mapping the interrelation of subsystems will provide necessary information about how the network functions as a whole. While many more examples can be discussed, in the following thesis a network will be approached in a general way: as a collection of N linear subsystems, which input/output (I/O) behaviour can be expressed in a state space representation:

$$x_{i}(t+1) = A_{i}x_{i}(t) + B_{i}v_{i}(t),$$

$$w_{i}(t) = C_{i}x_{i}(t).$$
(1)

Here, $x_i(t) \in \mathbb{R}^{n_i}$ is the state of the *i*-th subsystem, $v_i(t) \in \mathbb{R}^{m_i}$ is its input and $w_i(t) \in \mathbb{R}^{p_i}$ its output. The contribution of each node's output $w_i(t)$ to the external network output $y(t) \in \mathbb{R}^p$ is determined by the real $p \times p_i$ matrix S_i . The external network output is then defined as

$$y(t) = \sum_{i=1}^{N} S_i w(t).$$
 (2)

Due to influences between nodes, the node input $v_i(t)$ is composed of internal and external influences, and is defined as

$$v_i(t) = \sum_{j=1}^{N} Q_{ij} w_j(t) + R_i u(t),$$
(3)

where $u(t) \in \mathbb{R}^m$ is the external network input and Q_{ij} and R_i are real matrices of appropriate dimensions.

The matrix Q is the *interconnection matrix* of a network. The entry on the *i*-th row and *j*-th column, Q_{ij} , then describes the connection of node *j* with node *i*. More specifically, a nonzero value of Q_{ij} expresses an influence of the output of node *j* on the input of node *i*.

We continue by defining topology. Consider a directed graph $\mathcal{G}(Q) = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V} = 1, 2, \ldots, N$ corresponds to the nodes in the network. Then, a nonzero value of Q_{ij} corresponds to an edge from node j to node i, i.e. $(j, i) \in \mathcal{E}$ if and only if $Q_{ij} \neq 0$. The graph $\mathcal{G}(Q)$ is then referred to as the topology of a network. Then, the problem of topology identification concerns finding $\mathcal{G}(Q)$ (equivalently, interconnection matrix Q), using measurements of external network input u(t) and output y(t).

2.2 Problem owners and stakeholders

The main problem owner is PhD student Henk J. van Waarde, who is currently conducting research in the field of network identification for the University of Groningen. Furthermore, prof. Claudio de Persis is a stakeholder since he is one of the leading professors in the field of control engineering for the University of Groningen and the first supervisor of this thesis.

2.3 System and scope

2.3.1 System description

In the following section, a description of the system on which the research is focused will be given. First, the dynamics within a node will be analysed. Then, a system description of a network will be provided. Finally, the process of topology identification will be depicted in the last system.

Let us first consider the dynamics within a single node i, which are expressed in (1). Since the node is part of a network of systems, the input $v_i(t)$ is composed of internal and external influences, as expressed in (3). Then, the dynamics of a single node can be depicted in a block diagram, as can be seen in Figure 1.



Figure 1: Block diagram of single node dynamics.

Note that the inputs of this system are both the external input u(t) and the internal input $\sum_{j=1}^{N} Q_{ij} w_j(t)$, and the system output is the node output $w_i(t)$. In terms of efficacy, the previously introduced S_i matrix describes the contribution of the system to the higher system, which is the network.

We now consider a network of N nodes. The system is depicted in Figure 2.



Figure 2: Diagram of network dynamics.

In the network system, the system input is solely the external input u(t) and the system output is the external output y(t).

Finally, we consider the process of topology identification. First of all, we measure input/output (I/O) data from the network. Then we use this data as inputs to the topology identification algorithm. Finally, the output of the algorithm is the topology of the network. This process is depicted in Figure 3.



Figure 3: The process of topology identification.

2.3.2 Scope

In the following thesis, the network dynamics will be assumed to be linear and time-invariant. Other assumptions are:

- The network is considered to be heterogeneous, i.e. the A_i , B_i and C_i -matrices of all subsystems are not necessarily equal.
- The A_i , B_i and C_i -matrices of all subsystems are known.
- The network dynamics are in discrete-time.
- The in- and outputs of the network are measurable and there is no measurement noise.

2.4 Research goal

Consider the network dynamics as introduced in Section 2.1. The research goal of the following thesis is then to construct an algorithm that identifies the network topology based on measured input/output data, by using a subspace identification method.

2.5 Tools and methods

In this thesis, several tools and methods have been used. These are discussed below.

2.5.1 Literature research

Literature research has been conducted to investigate current knowledge on topology identification in order to construct the algorithm. Some preliminary literature results are discussed below:

- Control Theory for Linear Systems [8]. This book provides a clear overview of the background of control theory, and hence an early insight in the possibilities current theories may provide.
- On- and off-line identification of linear state space models [9]. This paper already provides an algorithm for identification of a single linear timeinvariant system. This algorithm can be applicable to an entire network if the network dynamics are expressed in an applicable format.

2.5.2 MatLab

The constructed algorithm has been coded in MatLab, a programming language in which the input/output data can be generated and be used as inputs to the algorithm. MatLab hence serves as a tool for simulation and validation of the algorithm.

2.5.3 Cycle choice

In order to analyze the design process in the following thesis, the design cycle approach formulated in [10] has been used. This theory makes a distinction between three cycles: the relevance cycle, the design cycle and the rigor cycle. These are depicted in Figure 4.



Figure 4: The design cycles as defined by Hevner.

In this thesis, the main focus will lie on the rigor cycle, since the construction of the algorithm will be based primarily on scientific theories and methods. In the literature research, already existing algorithms for topology identification (Meta-Artifacts) will be assessed and, if possible, used in the construction process. Furthermore, the final design goal is an addition to the current knowledge base. The final algorithm will be applicable in many different areas; topology identification in networks of linear systems is relevant in many fields, as explained in the introduction.

Since there is no direct application of the algorithm in for example a company, the relevance cycle is less applicable: defining clear requirements and conducting field experiments are not incorporated in the scope of this thesis.

Finally, the design cycle will have a minor influence on the process, since the final algorithm will serve as a new artifact that forms the output of the design cycle.

3 Results

In the next section, the found topology identification method is explained. First of all, in Section 3.1, a method is proposed to write the dynamics of a network of systems in a single state space representation. Secondly, in Section 3.2, the theory behind subspace identification is introduced as a method to determine the network dynamics based on this expression. Finally, in Section 3.3, a system of linear equations is derived, for which the results from subspace identification are used to solve the system to find the network topology.

3.1 Compact formulation of network dynamics

Consider a network of N linear time-invariate systems, where each system's dynamics are as expressed in (1). Let the node input $v_i(t)$, node output $w_i(t)$, node state $x_i(t)$, external input u(t) and external output y(t) be as defined in Section 2.1. Then, by introducing the block diagonal matrices

$$A = \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_N \end{pmatrix}, B = \begin{pmatrix} B_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B_N \end{pmatrix},$$

$$C = \begin{pmatrix} C_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_N \end{pmatrix},$$

and the block matrices

$$Q = \begin{pmatrix} Q_{11} & \dots & Q_{1N} \\ \vdots & \ddots & \vdots \\ Q_{N1} & \dots & Q_{NN} \end{pmatrix}, \ R = \begin{pmatrix} R_1 \\ \vdots \\ R_N \end{pmatrix}, \ S^\top = \begin{pmatrix} S_1^\top \\ \vdots \\ S_N^\top \end{pmatrix},$$

we can represent the entire network dynamics as

$$x(t+1) = (A + BQC)x(t) + BRu(t),$$

$$y(t) = SCx(t),$$
(4)

where the vector $x(t) \in \mathbb{R}^n$ is the concatenation of the states of all network nodes and $n = \sum_{i=1}^N n_i$.

We introduce new matrices E, F, G, where E = (A + BQC), F = BR and G = SC. The system can then be expressed compactly as

$$\begin{aligned} x(t+1) &= Ex(t) + Fu(t), \\ y(t) &= Gx(t), \end{aligned} \tag{5}$$

which corresponds to a general state space form.

3.2 Subspace identification

In the following section, the theory behind subspace identification is explained. Throughout this section, let A, B, C and D be general matrices defining the state-space system

$$x(t+1) = Ax(t) + Bu(t),
 y(t) = Cx(t) + Du(t).$$
(6)

Here, $u(t) \in \mathbb{R}^m$ denotes the system input, where m is the number of inputs, and $y(t) \in \mathbb{R}^p$ denotes the system output, where p is the number of outputs. Furthermore, $x(t) \in \mathbb{R}^n$ denotes the system state, where n denotes the number of states. Subspace identification involves finding matrices (A, B, C, D) based on I/O data.

3.2.1 Rewriting the model

The first step is to rewrite the model into an extended state space expression. Consider a sequence of inputs and outputs, (u(t), y(t), t = 0, 1, ..., j + i - 2), and a sequence of states, (x(t), t = 0, 1, ..., j-1), that satisfy (6). The sequence then satisfies [9]:

$$Y_h = \Gamma_i X + H_t U_h,\tag{7}$$

where Y_h is a block Hankel matrix containing consecutive outputs,

$$Y_h = \begin{pmatrix} y(0) & y(1) & \dots & y(j-1) \\ y(1) & y(2) & \dots & y(j) \\ \vdots & \vdots & \ddots & \vdots \\ y(i-1) & y(i) & \dots & y(j+i-2) \end{pmatrix}$$

Similarly, U_h is a block Hankel matrix with the same block dimensions as Y_h . This matrix contains consecutive inputs,

$$U_h = \begin{pmatrix} u(0) & u(1) & \dots & u(j-1) \\ u(1) & u(2) & \dots & u(j) \\ \vdots & \vdots & \ddots & \vdots \\ u(i-1) & u(i) & \dots & u(j+i-2) \end{pmatrix}.$$

Furthermore, X contains consecutive state vectors,

$$X = \begin{pmatrix} x(0) & x(1) & x(2) & \dots & x(j-1) \end{pmatrix},$$

 Γ_i is an extended observability matrix defined as

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix},$$

and H_t is a lower triangular block Toeplitz matrix,

$$H_t = \begin{pmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2} & CA^{i-3}B & CA^{i-4} & \dots & D \end{pmatrix}.$$

The expression in (7) is called the extended state space model. In this model, i and j are arbitrary integers. Let s denote the available amount of data samples. Guidelines for the values of i and j are then to have j typically equal to s-2i+1, which implies that all s available data samples are used, and i as at least the upper bound of the expected order of the system [11].

3.2.2 Matrix identification methods

Among all subspace identification techniques, a distinction is made between two different approaches to identify matrices (A, B, C, D): through the state vector sequence or the extended observability matrix [12]. The first one involves determining a vector state sequence

$$X = (x(0) \quad x(1) \quad x(2) \quad \dots \quad x(j-1)).$$

Once this is obtained, the matrices can be identified up to a similarity matrix T by solving the following system of linear equations:

$$\begin{pmatrix} x(1) & x(2) & \dots & x(j-1) \\ y(0) & y(1) & \dots & y(j-2) \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} x(0) & x(1) & \dots & x(j-2) \\ u(0) & u(1) & \dots & u(j-2) \end{pmatrix}.$$

In order to find a state vector sequence, several methods have been constructed. One of these will be elaborated on in section 3.2.3.

The second one first determines the extended observability matrix Γ_i . Then, the C matrix can be read directly from

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{pmatrix}.$$

Let Γ_i be the extended observability matrix with the last block row removed, and let $\underline{\Gamma}_i$ be the extended observability matrix with the first block row removed:

$$\bar{\Gamma}_i = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \end{pmatrix}, \quad \underline{\Gamma}_i = \begin{pmatrix} CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix}.$$

This implies

$$\Gamma_i A = \Gamma_i,$$

which is linear in A and can be solved to find this matrix.

While this method provides an efficient way of determining A and C, determining B and D using this method is more complex than through the state vector sequence. It has been found that identification of the B-matrix is required for topology identification. Therefore, subspace identification through the extended observability matrix will not be considered. In the next section, a technique to find the state vector sequence is proposed.

3.2.3 State vector sequence approximation

In order to construct the state vector sequence, a theory is proposed in [9]. First, the extended state space expression in (7) is divided into two expressions: one for the "past" and one for the "future". We make a distinction between Y_{h1} ("past" output values) and Y_{h2} ("future" output values), where

$$Y_{h1} = \begin{pmatrix} y(0) & y(1) & \dots & y(j-1) \\ y(1) & y(2) & \dots & y(j) \\ \vdots & \vdots & \ddots & \vdots \\ y(i-1) & y(i) & \dots & y(j+i-2) \end{pmatrix}$$

and

$$Y_{h2} = \begin{pmatrix} y(i) & y(i+1) & \dots & y(i+j-1) \\ y(i+1) & y(i+2) & \dots & y(i+j) \\ \vdots & \vdots & \ddots & \vdots \\ y(2i-1) & y(2i) & \dots & y(j+2i-2) \end{pmatrix}.$$

The "past" and "future" input matrices, respectively U_{h1} and U_{h2} , are constructed similarly. Finally, the "past' and "future" state vector sequences, respectively X_1 and X_2 , are defined as

$$X_1 = \begin{pmatrix} x(0) & x(1) & x(2) & \dots & x(j-1) \end{pmatrix}, X_2 = \begin{pmatrix} x(i) & x(i+1) & x(i+2) & \dots & x(i+j-1) \end{pmatrix}.$$

This then yields the two I/O equations that satisfy (6):

$$Y_{h1} = \Gamma_i X_1 + H_t U_{h1}, \tag{8}$$

$$Y_{h2} = \Gamma_i X_2 + H_t U_{h2}.$$
 (9)

Now, before the state vector sequence can be determined, we introduce three conditions. Let Y_h , U_h and X be defined as in section 3.2.1, and let H denote the concatenation of Y_h and U_h :

$$H = \begin{pmatrix} Y_h \\ U_h \end{pmatrix}.$$

Furthermore, let the row space of a matrix A be denoted as rs(A). We then consider the conditions:

- 1. $\operatorname{rank}(X) = n$, i.e. all modes are sufficiently excited.
- 2. $rs(X) \cap rs(U_h) = \{0\}.$
- 3. $\operatorname{rank}(U_h) = mi = \operatorname{number} of rows in U_h$.

These conditions are necessary to estimate the system order n based on the I/O data, which is proven and further explained in [9]. Alternatively, it has been shown in [13] that a sufficient condition for system order estimation is that the input is persistently exciting of sufficiently high order. Now, let H_1 and H_2 be the concatenation of (Y_{h1}, U_{h1}) and (Y_{h2}, U_{h2}) respectively:

$$H_1 = \begin{pmatrix} Y_{h1} \\ U_{h1} \end{pmatrix}, \ H_2 = \begin{pmatrix} Y_{h2} \\ U_{h2} \end{pmatrix}.$$
 (10)

Then, if both (H_1, X_1) and (H_2, X_2) satisfy the conditions defined before, the row space of the state vector sequence X_2 can be determined by

$$\operatorname{rs}(X_2) = \operatorname{rs}(H_1) \cap \operatorname{rs}(H_2).$$
(11)

In order to determine this intersection, various methods exist. A first way, presented in [9], is by making use of a singular value decomposition of the concatenated Hankel matrix H. Alternatively, another method, presented in [14], involves taking as a basis for the intersection the principal directions between the row space of the past inputs and outputs and the row space of the future inputs and outputs.

The first method was coded into MatLab and then used to validate the final algorithm. This will be further elaborated on in the validation section.

3.2.4 Similarity transformation

Let (A, B, C, D) be the real matrices of a system. The solution of a subspace identification problem are then matrices $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, where

$$\bar{A} = T^{-1}AT, \ \bar{B} = T^{-1}B, \ \bar{C} = CT, \ \bar{D} = D,$$

for an unique invertible transformation matrix T. This is because a similar system has the same I/O behaviour, which can be seen by comparing the transfer function $\bar{H}(z)$ of $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ with the real transfer function H(z):

$$\bar{H}(z) = CT(zI - T^{-1}AT)^{-1}T^{-1}B + D = C(zI - A)^{-1}B + D = H(z).$$

3.3 Implementation in topology identification

In the following section, the previously described theory on subspace identification will be applied on network dynamics. First, the application of subspace identification on the network state space model in (5) is discussed. Then, the applications in topology identification will be evaluated.

By using subspace identification, the real system matrices (E, F, G) defined in Section 3.1 can be estimated up to a similarity transformation T. Let $(\bar{E}, \bar{F}, \bar{G})$ be the results of subspace identification algorithm. We then have that

$$\bar{E} = T^{-1}ET, \ \bar{F} = T^{-1}F, \ \bar{G} = GT,$$

for some unique invertible transformation matrix T. We assume that the individual dynamics of each subsystem are known (and hence matrices A, B, C, R, S are known), where S = R = I. We then get

$$\bar{E} = T^{-1}ET = T^{-1}(A + BQC)T,$$
(12)

$$\bar{F} = T^{-1}F = T^{-1}B,\tag{13}$$

$$\bar{G} = GT = CT. \tag{14}$$

Then, by substituting (14) into (12), we get

$$\bar{E} = T^{-1}AT + T^{-1}BQ\bar{G}.$$

Finally, pre-multiplying the resulting equation with T yields

$$T\bar{E} = AT + BQ\bar{G}.$$
(15)

As a result, we have a system of linear equations in the unknowns T and Q:

$$T\bar{E} - AT = BQ\bar{G},$$

$$CT = \bar{G},$$

$$T\bar{F} = B.$$
(16)

In order to convert this system into a form that can be solved using standard techniques, we use the Kronecker product. Let the Kronecker product between $m \times n$ matrix X and $p \times q$ matrix Y, denoted as $X \otimes Y$, be the $mp \times nq$ block matrix:

$$X \otimes Y = \begin{pmatrix} x_{11}Y & \dots & x_{1n}Y \\ \vdots & \ddots & \vdots \\ x_{m1}Y & \dots & x_{mn}Y \end{pmatrix}.$$

Furthermore, let the vectorization of an arbitrary $n \times m$ matrix Z be denoted as $\operatorname{vec}(Z)$, where $\operatorname{vec}(Z) \in \mathbb{R}^{mn}$. We then use that the vectorization of the product of three arbitrary matrices K, L, M, of appropriate dimensions, can be rewritten as

$$\operatorname{vec}(KLM) = (M^{\top} \otimes K)\operatorname{vec}(L).$$
(17)

Then, vectorizing the equations in (16) and using the new expression obtained in (17) yields

$$\begin{split} (\bar{E}^{\top} \otimes I) \operatorname{vec}(T) &- (I \otimes A) \operatorname{vec}(T) = (\bar{G}^{\top} \otimes B) \operatorname{vec}(Q), \\ (I \otimes C) \operatorname{vec}(T) &= \operatorname{vec}(\bar{G}), \\ (\bar{F}^{\top} \otimes I) \operatorname{vec}(T) &= \operatorname{vec}(B), \end{split}$$

which can be expressed as

$$\begin{pmatrix} \bar{E}^{\top} \otimes I - I \otimes A & -(\bar{G}^{\top} \otimes B) \\ I \otimes C & 0 \\ \bar{F}^{\top} \otimes I & 0 \end{pmatrix} \begin{pmatrix} \operatorname{vec}(T) \\ \operatorname{vec}(Q) \end{pmatrix} = \begin{pmatrix} 0 \\ \operatorname{vec}(\bar{G}) \\ \operatorname{vec}(B) \end{pmatrix}.$$
(18)

This system of linear equations can be solved using standard methods, for example using the "backslash" command in Matlab. It can be shown that the solution to (18) is unique if so-called *identifiability conditions* on Q are satisfied. Such conditions are rather involved, and hence considered out of the scope of this thesis.

4 Validation

In order to validate the method discussed in this thesis, the topology identification procedure was coded into MatLab. The code can be found in appendices A.1 and A.2. In the following section, a description of the MatLab code is provided together with the validation procedure.

Consider a network of 5 nodes, and let the associated graph \mathcal{G} be as depicted in Figure 5. From the graph, the network's topology can immediately be seen by looking at the edges between the nodes.



Figure 5: A graph of a network of 5 nodes and their interconnections.

We let the interconnection matrix Q of the network be equal to the Laplacian matrix L of \mathcal{G} :

$$Q = L = \begin{pmatrix} 2 & 0 & -1 & 0 & -1 \\ 0 & 2 & -1 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Furthermore, individual node dynamics, i.e. the (A_i, B_i, C_i) matrices, are defined for the network. For illustration purposes, we let $n_i = 2$ and $m_i = p_i = 1$ for all nodes. The matrices are given by:

$$A_{1} = \begin{pmatrix} 0.1 & 0.3 \\ 0.1 & 0.4 \end{pmatrix}, A_{2} = \begin{pmatrix} 0.2 & 0.4 \\ 0.2 & 0.1 \end{pmatrix}, A_{3} = \begin{pmatrix} 0.4 & 0.1 \\ 0.2 & 0.1 \end{pmatrix}, A_{4} = \begin{pmatrix} 0.5 & 0.1 \\ 0 & 0.1 \end{pmatrix},$$
$$A_{5} = \begin{pmatrix} 0 & 0.1 \\ 0.1 & 0.5 \end{pmatrix},$$
$$B_{1} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, B_{2} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, B_{3} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, B_{4} = \begin{pmatrix} 0 \\ 1.0 \end{pmatrix}, B_{5} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$
$$C_{1} = \begin{pmatrix} 0.1 & 0 \end{pmatrix}, C_{2} = \begin{pmatrix} 0 & 0.1 \end{pmatrix}, C_{3} = \begin{pmatrix} 0.2 & 0.1 \end{pmatrix}, C_{4} = \begin{pmatrix} 0.3 & 0 \end{pmatrix},$$
$$C_{5} = \begin{pmatrix} 0.1 & 0.1 \end{pmatrix}.$$

Based on these, matrices A, B and C are constructed using the "blkdiag" command in MatLab. Then, in order to run the method, I/O data is generated for the network based on the network dynamics. This can be seen in Appendix A.1.

Then, in the code in Appendix A.2, matrices \overline{E} , \overline{F} , \overline{G} are determined by running the algorithm proposed in [9]. We highlight matrices \bar{F} and \bar{G} resulting from the algorithm:

$$\bar{G} = \begin{pmatrix} -0.17 & 0.00 & 0.06 & 0.00 & 0.28 \\ -0.18 & -0.04 & 0.03 & 0.00 & -0.15 \\ -0.06 & 0.04 & -0.12 & 0.00 & 0.05 \\ 0.03 & -0.11 & -0.03 & 0.00 & 0.07 \\ -0.02 & -0.01 & 0.00 & 0.03 & -0.02 \\ 0.00 & 0.00 & 0.00 & 0.02 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.01 & -0.02 \\ -0.01 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.01 & -0.02 & -0.67 & -0.27 & 0.04 & -0.02 & -0.01 & 0.19 & -0.34 & 0.55 \\ 0.01 & -0.07 & -0.02 & -0.06 & 0.56 & -0.81 & 0.10 & -0.07 & -0.04 & -0.12 \\ 0.62 & -0.58 & 0.10 & 0.26 & -0.09 & -0.11 & -0.57 & 0.19 & 0.00 & -0.02 \end{pmatrix}$$

0.00

-0.02

As expected, the matrices B and C are not equal to F and G, respectively, since the subspace identification method identifies the matrices up to a transformation matrix T.

Finally, the analytical tools proposed in Section 3.3 were implemented to find the resulting matrices T^* and Q^* , the code of which is in Appendix A.1. The resulting matrix Q^* is then

$$Q^* = \begin{pmatrix} 2 & 0 & -1 & 0 & -1 \\ 0 & 2 & -1 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

which is equal to the matrix Q. Based on this matrix, we can reconstruct the graph depicted in Figure 5. We conclude that the method is successful.

5 Conclusion

We conclude with the words of E.O. Wilson [15]: "The greatest challenge today, not just in cell biology and ecology but in all of science, is the accurate and complete description of complex systems. Scientists have broken down many kinds of systems. They think they know most of the elements and forces. The next task is to reassemble them, at least in mathematical models that capture the key properties of the entire ensembles."

In this thesis, it has been found that an existing method of subspace identification can be applied to identify the dynamics of a network. Furthermore, these results can then be used to solve a system of linear equations for the interconnection matrix Q, from which the network's topology can be derived. Finally, the method has been coded in Matlab and it was validated based on a network of 5 nodes.

It can be concluded that, under suitable identifiability conditions, the approach proposed in this thesis can successfully identify a network's topology, using a subspace identification method. Due to the general network approach, it could be applicable in many disciplines. However, in order to convert the method into an applicable format for real life situations, much future research should be conducted.

6 Discussion and further research

In the following section, several topics left for discussion and/or future research are mentioned:

• In this thesis, several restricting assumptions have been made. For future research, the influence of relaxations of these assumptions has to be assessed in order to make the model more applicable in the real world. These assumptions are:

- Linearity of the subsystems. In real life complex networks, linearity often is not a realistic assumption. Therefore, for future research, linearization methods for nonlinear systems can be looked into. Subspace identification of non-linear systems has already been researched in current literature, such as identification of Wiener systems [16] or Hammerstein systems [17].
- Time-invariance of the network. Similar to linearity, the assumption of time-invariance is a strong one when trying to approach real-life behaviour. Not only node dynamics can change over time; the topology can be time-varying as well. A simple solution would be to repeatedly run the algorithm to re-determine the network topology. Alternatively, synchronization-based topology identification methods have been constructed for networks with time-variant behaviour [18].
- No feedthrough. The expression proposed in section 3.2.1 can be adjusted to incorporate the feedthrough matrices for each system. Furthermore, the feedthrough matrix can be obtained by means of the subspace identification algorithm. This inclusion will be left to further research.
- No measurement noise. When measurement noise occurs, there is likely not an exact solution to the linear equations in sections 3.2.2 and 3.3. Therefore, a least-squares method can be used to solve the equations and hence find an approximation for the network topology.
- Discrete-time network dynamics. Although we focus on discretetime systems, our results can be stated for continuous-time systems as well. By introducing a subspace identification algorithm that functions in continuous time, e.g. the method proposed in [19], the matrices $\bar{E}, \bar{F}, \bar{G}$ can be determined similarly and hence be used to determine a network's topology.
- The system of linear equations in (18) is computationally demanding, especially when large networks are considered. Alternatively, the equation in (15) can be treated as a generalized Sylvester equation. By introducing matrix $W = Q\bar{G}$, we get an equation of the form AV - EVF = BWin the unknowns V and W. In [20], a method is discussed to solve this equation for V and W up to an arbitrary parameter matrix Z, under suitable conditions on matrices E, A and B. By using this alternative method, it could be that the computational load of solving the system of linear equations is reduced. This will be left to future research.

References

 J. Zhou and J. Lu. Topology identification of weighted complex dynamical networks. *Physica A: Statistical Mechanics and its Applications*, 386(1):481–491, 2007.

- [2] J. Zhao, Q. Li, J. Lu, and Z. Jiang. Topology identification of complex dynamical networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(2): Article 023119, 2010.
- [3] W. K. S. Tang, M. Yu, and L. Kocarev. Identification and monitoring of biological neural network. In 2007 IEEE International Symposium on Circuits and Systems, pages 2646–2649, 2007.
- [4] M. Nabi-Abdolyousefi and M. Mesbahi. Network identification via node knockout. *IEEE Transactions on Automatic Control*, 57(12):3214–3219, 2012.
- [5] M. Suzuki, N. Takatsuki, J. Imura, and K. Aihara. Node knock-out based structure identification in networks of identical multi-dimensional subsystems. In 2013 European Control Conference (ECC), pages 2280–2285, 2013.
- [6] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal* Society B: Biological Sciences, 360(1457):969–981, 2005.
- [7] S. Nabavi and A. Chakrabortty. Topology identification for dynamic equivalent models of large power system networks. In 2013 American Control Conference, pages 1138–1143. 2013.
- [8] Harry L. Trentelman, Antonie A. Stoorvogel, and Malo Hautus. Control Theory for Linear Systems. Communications and Control Engineering. Springer, 2001.
- [9] Marc Moonen, Bart de Moor, Lieven Vandenberghe, and Joos Vandewalle. On- and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232, 1989.
- [10] Alan Hevner. A three cycle view of design science research. Scandinavian Journal of Information Systems, 19: Article 4, 2007.
- [11] Katrien De Cock and Bart De Moor. Subspace identification. In EOLSS, UNESCO Encyclopedia of Life Support Systems, pages 933–979. 2003.
- [12] Wouter Favoreel, Bart De Moor, and Peter Van Overschee. Subspace state space system identification for industrial processes. *Journal of Process Control*, 10(2):149–155, 2000.
- [13] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L.M. De Moor. A note on persistency of excitation. Systems Control Letters, 54(4):325– 329, 2005.

- [14] Bart de Moor, Marc Moonen, Lieven Vandenberghe, and Joos Vandewalle. The application of the canonical correlation concept to the identification of linear state space models. *Lecture Notes in Control and Information Sciences*, 111:1103–1114, 2006.
- [15] Edward O. Wilson. Consilience and complexity. Complexity, 3(5):17–21, 1998.
- [16] Michel Verhaegen and David Westwick. Identifying mimo wiener systems using subspace model identification methods. *Signal Processing*, 52:4206– 4211, 1996.
- [17] Michel Verhaegen and David Westwick. Identifying mimo hammerstein systems in the context of subspace model identification methods. *IFAC Proceedings Volumes*, 29(1):4092–4097, 1996.
- [18] Hai-Peng Ren, Kun Tian, and Ren Zhou. Local time-varying topology identification of network with unknown parameters based on adaptive synchronization. Automatika, 59(3-4):391–399, 2018.
- [19] Zi-Jiang Yang, Setsuo Sagara, and Kiyoshi Wada. Subspace model identification for continuous-time systems. *IFAC Proceedings Volumes*, 30(11):1567–1572, 1997.
- [20] Bin Zhou and Guang-Ren Duan. A new solution to the generalized Sylvester matrix equation AV-EVF=BW. Systems & Control Letters, 55(3):193–198, 2006.

Appendices

A Matlab codes

A.1 MatLab code of topology identification

```
clear all
%% Determine matrices
%Determine matrices of each subsystem
A1 = [0.1 0.3;0.1 0.4];
A2 = [0.2 0.4;0.2 0.1];
A3 = [0.4 0.1;0.2 0.1];
A4 = [0.5 0.1;0 0.1];
A5 = [0 0.1;0.1 0.5];
B1 = [2;1];
B2 = [1;1];
B3 = [0;1];
```

```
B4 = [0;1];
B5 = [2;1];
C1 = [0.1 \ 0];
C2 = [0 \ 0.1];
C3 = [0.2 \ 0.1];
C4 = [0.3 0];
C5 = [0.1 \ 0.1];
%Determine the Q-matrix
Q = [2 \ 0 \ -1 \ 0 \ -1; 0 \ 2 \ -1 \ -1 \ 0; -1 \ -1 \ 3 \ -1 \ 0; 0 \ -1 \ -1 \ 2
   0; -1 \ 0 \ 0 \ 1];
%Construct the A,B,C matrices for network notation
A = blkdiag(A1, A2, A3, A4, A5);
B = blkdiag(B1,B2,B3,B4,B5);
C = blkdiag(C1, C2, C3, C4, C5);
%% Generate data
%Determine number of states, inputs and outputs
[n,~] = size(A);
[~,m] = size(B);
[p,~] = size(C);
%Generate random input data and initial state
T = 1000;
x = rand(n, 1);
U = rand(m,T);
Y = zeros(p,T);
for i = 1:T
    Y(:,i) = C * x;
    x = (A+B*Q*C)*x + B*U(:,i);
end
%% Topology identification
%Run the subspace identification algorithm
[E,F,G,H] = subspace_id_p(n,U,Y);
%Prepare matrices to set up the system of linear
   equations
I=eye(n);
E_kron_I = kron(E',I);
I_kron_A = kron(I,A);
```

```
G_kron_B = kron(G',B);
I_kron_C = kron(I,C);
F_kron_I = kron(F',I);
vec_G = G(:);
vec_B= B(:);
z_left=zeros(n*p,p*m);
z_right=zeros(n<sup>2</sup>,1);
Y_11=E_kron_I-I_kron_A;
Y=[Y_11 -G_kron_B; I_kron_C z_left; F_kron_I z_left];
Z=[z_right; vec_G; vec_B];
\% \, \text{Solve} the system of linear equations to find the T-
   and Q-matrix
X = Y \setminus Z;
T_vec = X(1:n^2, 1);
Q_vec = X(n^2+1:n^2+p*m);
T_ans=reshape(T_vec,n,n);
Q_ans=reshape(Q_vec,m,p);
```

A.2 MatLab code of subspace identification algorithm

```
function [E,F,G,H] = subspace_id_p(n,u,y)
\%Choosing i and j and determining the number of in-
   and outputs
%settings
[~,s]=size(u);
i=n;
j=s-2*i+1;
%number of inputs
sz_m=size(u);
m = sz_m(1);
%number of outputs
sz_l=size(y);
p=sz_l(1);
\%\% Determining the H matrix based on the input and
   output
%Preparation
H_1=zeros((m*i+p*i),j);
H_2=zeros(i,j);
%Assiging data to H_1 matrix
for k = 1:i
    H_1((1+(k-1)*(m+p)):(1+(k-1)*(m+p)+m-1),1:j)=u(1:m)
       ,k:k+j-1);
    H_1((m+1+(k-1)*(m+p)):(k*(m+p)),1:j) = y(1:p,k:k+j)
       -1);
end
%Assigning data to H_2 matrix
for k = 1:i
    H_2((1+(k-1)*(m+p)):(1+(k-1)*(m+p)+m-1),1:j)=u(1:m)
       ,k+i:k+i+j-1);
    H_2((m+1+(k-1)*(m+p)):(k*(m+p)), 1:j) = y(1:p,k+i:k)
       +i+j-1);
end
%Finally the H matrix is given by
H = [H_1; H_2];
%% Starting the algorithm
%Computing SVD of H
[U,S,V] = svd(H);
```

```
\% Separating the relevant matrices U_11, U_12 and S_11
U_11=U(1:(m*i+p*i),1:(2*m*i+n));
U_12=U(1:(m*i+p*i),(2*m*i+n+1):(2*m*i+2*p*i));
S_11=S(1:(2*m*i+n),1:(2*m*i+n));
%find the second SVD
U_12t=transpose(U_12);
K=U_{12t}*U_{11}*S_{11};
[U2, S2, V2] = svd(K);
%find the finally required matrices
U_q=U2(1:(2*p*i-n),1:n);
U_qt=transpose(U_q);
M_1=U_qt*U_12t*U(m+p+1:(i+1)*(m+p),:)*S;
M_2=U(m*i+p*i+m+1:(m+p)*(i+1),:)*S;
M_3=U_qt*U_12t*U(1:m*i+p*i,:)*S;
M_4=U(m*i+p*i+1:m*i+p*i+m,:)*S;
M_1 = [M_1; M_2];
M_r = [M_3; M_4];
size(M_l);
size(M_r);
%% solve the set of equations
X = M_r' \setminus M_1';
X = X';
E = X(1:n, 1:n);
```

F = X(1:n,n+1:n+m); G = X(n+1:n+p,1:n); H = X(n+1:n+p,n+1:n+m);

```
23
```