Identifiability and Identification Methods for Dynamic Networks

Citation for published version (APA):
Weerts, H. H. M. (2018). *Identifiability and Identification Methods for Dynamic Networks*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven.

Document status and date:
Published: 07/11/2018

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

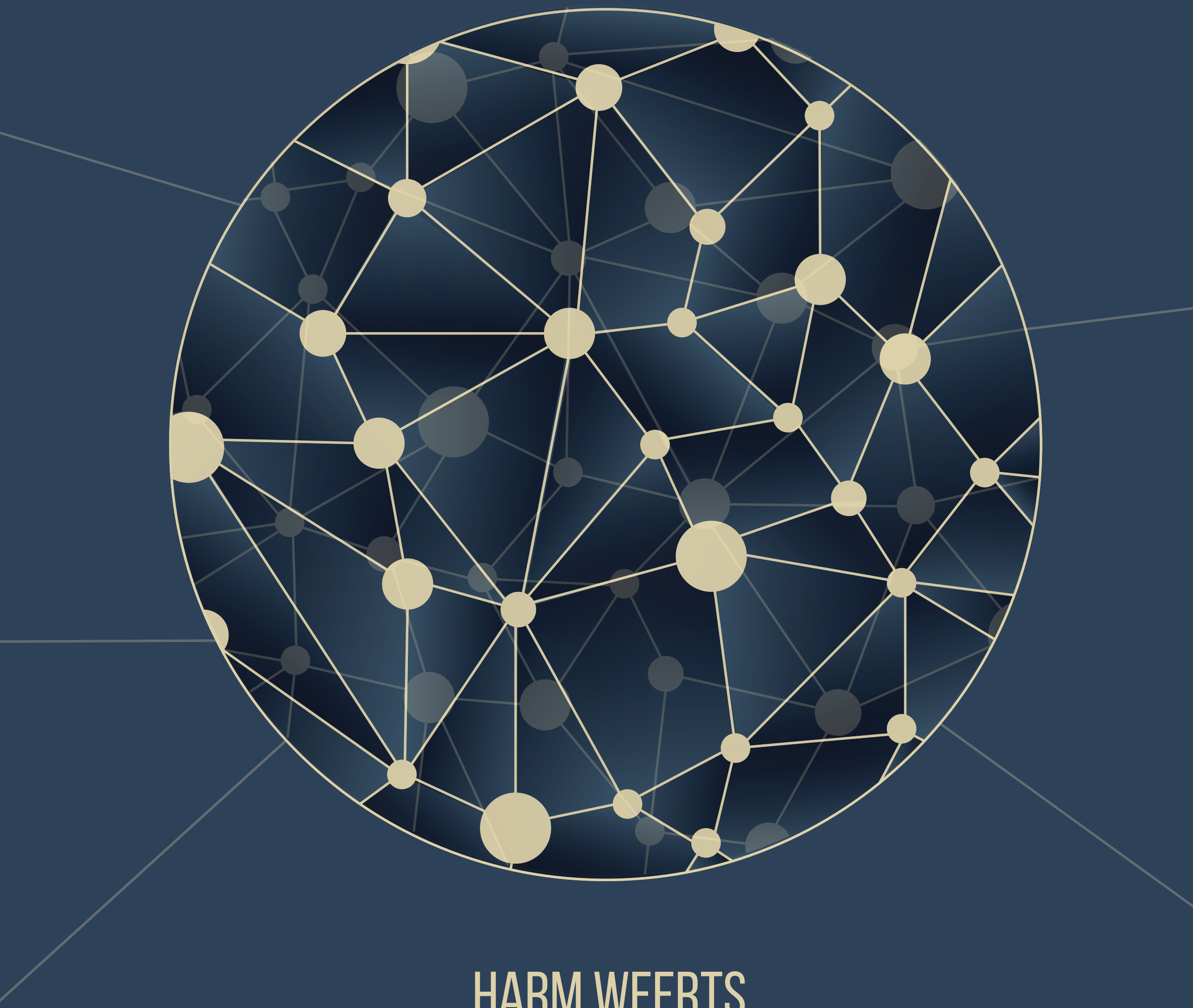
Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 05. Oct. 2023

Identifiability and Identification Methods for Dynamic Networks



HARM WEERTS

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Identifiability and identification methods for dynamic networks

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op woensdag 7 november 2018 om 13:30 uur

door

Hermanus Henricus Maria Weerts

geboren te Bergen (L)

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	Prof. dr. ir. J. H. Blom
promotor:	Prof. dr. ir. P. M. J. Van den Hof
co-promotor:	dr. A. G. Dankers MSc (University of Calgary)
leden:	Prof. dr. J. Gonçalves (University of Luxemburg)
	Prof. dr. ir. J. M. A. Scherpen (RU Groningen)
	Prof. dr. ir. N. A. W. van Riel
	Prof. dr. S. Weiland
	Prof. dr. M. Enqvist (Linköping University)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Identifiability and identification methods for dynamic networks

Harm H.M. Weerts

This project has received funding from the European Research Council (ERC), Advanced Research Grant SYSDYNET, under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 694504).

The research reported in this thesis is part of the research program of the Dutch Institute of Systems and Control (DISC). The author has successfully completed the educational program of the Graduate School DISC.

Identifiability and identification methods for dynamic networks
by Hermanus Henricus Maria Weerts
Eindhoven, Technische Universiteit Eindhoven, 2018. Proefschrift.

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-4624-4

This thesis was prepared with the LaTeXdocumentation system.
Cover design: Ingrid Weerts
Reproduction: Ipskamp Printing, Enschede, The Netherlands

Copyright © 2018 by H.H.M. Weerts

Samenvatting

Identifiability and identification methods for dynamic networks

Opwarming van de aarde zal gelimiteerd worden door de uitstoot van broeikasgassen te verminderen. Hierin speelt verdere ontwikkeling van technologiën zoals hernieuwbare energiebronnen en een slim electriciteitsnet een hoofdrol. Sommige uitdagingen in het slimme electriciteitsnet zoals oscillaties, instabilitiet, en het ontwerp van regelaars kunnen worden opgelost met behulp van dynamische modellen. Het electriciteitsnet en andere moderne toepassingen bestaan uit vele gekoppelde sub-systemen, en de interacties tussen sub-systemen kunnen worden weergegeven in een *dynamisch netwerk model*. Een dynamisch netwerk model bestaat uit knooppunten die gemeten signalen weergeven, modules die het dynamische gedrag weergeven, en excitatie vanuit de niet-gemodelleerde wereld. Modelleren op basis van data is een manier om een dynamisch netwerk model te verkrijgen, en het veld bekend als systeem identificatie houdt zich bezig met het modelleren van dynamische systemen. Het kader van dit proefschrift is gelimiteerd tot het identificeren van lineare tijd-invariante netwerken op basis van data.

Identificatie methodes worden gewoonlijk opgesteld op basis van een beperkte experimentele structuur. In het geval van een dynamisch netwerk wordt bijvoorbeeld een beperkte topologie opgelegd, wat kan leiden tot een onnauwkeurig model wanneer de verkeerde topologie gekozen is. Daarbij wordt gekozen welke knooppunten gemeten worden, maar het is niet exact duidelijk welke knooppunten gemeten moeten worden om een bepaald doel te bereiken. Een flexibele experimentele structuur wordt opgezet waar verschillende keuzes voor topologie en externe excitatie mogelijk zijn. Het uitbreiden van netwerk identificatie methodes voor deze flexibele experimentele structuur is niet triviaal. In dit proefschrift wordt vastgesteld welke restricties in de experimentele structuur voldoende en noodzakelijk zijn, en een efficiënte identificatie methode voor deze flexibele structuur wordt opgesteld zowel in theorie als in een functioneel algoritme.

Met een flexibele experimentele structuur kan het voorkomen dat twee netwerk modellen niet kunnen worden onderscheiden op basis van data. Het concept *netwerk identificeerbaarheid* wordt geïntroduceerd als eigenschap die beschrijft of dat modellen van elkaar onderscheiden kunnen worden. Netwerk identificeerbaarheid kan worden

gegarandeerd door beperkingen in the experimentele structuur aan te brengen, bijvoorbeeld door de toegestane dynamica, topologie, externe excitaties, of verstoringen te beperken. Eenvoudig te verifiëren condities op basis van de netwerk topologie zijn opgesteld die garanderen dat een set van netwerk modellen generiek netwerk identificeerbaar is. Op deze manier kan de meest flexibele experimentele structuur gebruikt worden om te modelleren.

Het is een uitdaging om dynamische netwerk modellen van hoge kwaliteit te schatten voor situaties met algemene verstoringen. Gecorreleerde verstoringen met een spectrum van gereduceerde rang worden nauwelijks geadresseerd in de literatuur. De *joint-direct methode* wordt geïntroduceerd als een oplossing voor deze uitdaging. Een afleiding op basis van een analyse van voorspellers leidt naar deze joint-direct methode. De methode leidt tot schattingen van hoge kwaliteit, wat zichtbaar is in de bewezen eigenschappen. Onder bepaalde voorwaarden is de joint-direct methode gelijk aan de Maximum Likelihood schatting. Daarnaast kan de methode een variantie behalen die gelijk is aan de Crámer-Rao ondergrens. De Maximum Likelihood en Crámer-Rao ondergrens resultaten zijn opgesteld voor verstoringen met spectrum van gereduceerde rang. Wanneer de juiste weging wordt gekozen levert de joint-direct methode een efficiënte schatting.

Dynamische netwerk modellen kunnen opgesteld worden voor situaties waar fysieke grootheden direct gekoppeld zijn, zodat algebraïsche lussen in het netwerk ontstaan. Een traditionele voorspeller leidt tot onzuivere schattingen. Een analyze van verschillende voorspeller definities laat zien waarom de directe methode tot onzuivere schattingen leidt. De joint-direct methode gebruikt een andere voorspeller dan de directe methode, en dit leidt tot consistente schattingen, ook onder aanwezigheid van algebraïsche lussen. In vergelijking met methoden gebaseerd op projecties heeft de joint-direct methode minder strikte eisen aan externe excitaties, en een kleinere variantie. Algebraïsche lussen zijn geen probleem voor identificatie wanneer deze op een juiste manier aangepakt worden.

Om een enkele module te schatten in een netwerk is het voldoende om enkele knooppunten rondom de module te meten. Het is niet exact bekend welke set van knooppunten tot een nauwkeurige schatting leidt. Er zijn verschillende oplossingen voor dit vraagstuk, maar in dit proefschrift is gekozen om ongemeten knooppunten te verwijderen uit het netwerk model. Het *abstracte netwerk* is ontwikkeld, en dit netwerk wordt opgebouwd uit een gekozen aantal knooppunten terwijl het gedrag van de overgebleven knooppunten gelijk blijft. Onder bepaalde voorwaarden blijft de module gelijk in het abstracte netwerk, zodat de module geïdentificeerd kan worden met de gekozen knooppunten. De condities kunnen geverifieerd worden op basis van de originele netwerk topologie, zonder het abstracte netwerk af te leiden. Op deze manier zijn eenvoudige regels opgesteld om knooppunten te selecteren waarmee een specifieke module geïdentificeerd kan worden.

Krachtige algoritmes zijn nodig om de joint-direct methode efficiënt in de praktijk toe te passen op dynamische netwerken van grote schaal. Een *Sequential Least Squares* algoritme met expliciete oplossing is opgesteld op basis van een benadering van de joint-direct methode. Dit algoritme heeft dezelfde asymptotische eigenschappen als de joint-direct methode, en simulaties laten zien dat het algoritme naar dezelfde nauwkeurigheid convergeert als de joint-direct methode wanneer de hoeveelheid data groeit. Op deze manier is een efficiënte implementatie van en netwerk identificatie methode gerealizeerd.

Een theoretische basis voor de identificatie van dynamische netwerken is gelegd, en daarnaast zijn stappen genomen in de richting van identificatie van een module in een netwerk. Mogelijke vervolgstappen zijn een onderzoek naar de informativiteits eisen aan data, identificatie van de netwerk topologie, of het toevoegen van niet-lineare dynamica.

Summary

Identifiability and identification methods for dynamic networks

Global warming is to be limited by reducing greenhouse gas emissions. Further development of technologies such as renewable energy sources and smart electricity grids is to play a major role in this reduction. Some of the challenges in the smart grid such as oscillations, instability and control design may be solved with the help of a dynamic model. These smart grids and other modern applications consist of many interacting sub-systems, and the interactions between these sub-systems can be modeled as a *dynamic network model*. A dynamic network model consists of nodes that represent measured signals, modules that capture the dynamical behavior, and excitations coming from the unmodeled world. Data-driven modeling is one way of obtaining dynamic network models, and the field of modeling dynamical systems is known as system identification. The scope of this thesis is limited to the identification of linear time invariant networks.

Identification approaches are typically formulated on the basis of a restricted experimental setup. In the dynamic network situation for example the allowed network topology is restricted, which may lead to an inaccurate model if the wrong structure is chosen. Moreover a certain set of measured nodes is chosen for the experimental setup, but it is not clear which nodes need to be measured for a particular purpose. A flexible network setup is introduced where all different choices can be made for the topology and excitation locations. Extending network identification approaches to deal with a fully flexible experimental setup is not a trivial task. In this thesis it is to be determined which restrictions on the experimental setup are necessary and sufficient, and an efficient identification method for this setup is to be determined both in theory and as a functional algorithm.

With a fully flexible experimental setup it may happen that two network models can not be distinguished on the basis of data. The notion of *network identifiability* is introduced as the property that describes whether the models can be distinguished from each other. Network identifiability can be guaranteed by restricting the experimental setup, i.e. by restricting the modeled dynamics, topology, external inputs and disturbances. Easy to check conditions on the basis of the network topology are formulated

that guarantee that a network model set is generically network identifiable. In this way the least restrictive experimental setup can be used in the modeling process.

It is challenging to estimate high quality dynamic network models for general unknown disturbance situations. In particular correlated disturbances with a rank-reduced spectrum are rarely treated in literature. The *joint-direct method* is established as a solution for this problem. A derivation on the basis of an analysis of predictors leads to this joint-direct method. This joint-direct method leads to estimates of high quality. Under some conditions the joint-direct method is equivalent to the Maximum Likelihood estimate. Moreover, variance of models estimated with the joint-direct method is bounded from below by the Crámer-Rao Lower Bound. The Maximum Likelihood and Crámer-Rao Lower Bound results have been formulated for disturbances with rank-reduced spectrum. The joint-direct method leads to an efficient estimate when an appropriate weighting is chosen in the optimization cost function.

Dynamic network models may be formulated for situations where physical variables have direct couplings, such that algebraic loops appear in the network. A traditional direct estimation method leads to biased estimates. An analysis of different predictor definitions shows why the direct method leads to biased estimate. The joint-direct method uses a different predictor than the direct method, and leads to consistent estimates, even in the situation of algebraic loops. Compared to projection based methods, the joint-direct method has reduced requirements on external excitation signals, and leads to a reduction of variance. Algebraic loops are not a problem in the identification of systems when appropriately dealt with in the estimation method.

In order to estimate a single module in a network it is sufficient to measure a number of nodes locally around the module. It is however not precisely known which subset of nodes can lead to an accurate estimate. Answering this question can be done in different ways, and the approach taken is to remove unmeasured nodes from the network. The *abstracted network* is introduced, and this network can be constructed on the basis of a chosen set of nodes, and this leaves the behavior of the remaining nodes intact. Under certain conditions the module of interest remains invariant in the abstracted network, in which case it can be identified with the selected set of nodes. These conditions can be verified on the basis of the original network topology, without the need to construct the abstracted network. In this way straightforward rules for selecting a set of nodes are formulated that allow for identification of a module of interest.

Powerful algorithms are needed in order to efficiently apply the joint-direct method to large-scale dynamic networks in practice. A *Sequential Least Squares* algorithm with an explicit solution is formulated as an approximation of the joint-direct method. This algorithm has the same asymptotic properties as the joint-direct method, and simulations show that the algorithm converges to the same accuracy as the joint-direct method with increasing amounts of data. In this way an efficient implementation of network identification methods is formulated.

A theoretical basis for identification of dynamic networks has been established, and steps have been taken towards identification of a single module in a network. The next steps that can be taken are an investigation of informativity requirements on data, identification of the network topology, and the inclusion of non-linear dynamics.

Contents

1 Introduction

1.1 Motivation

1.1.1 Greenhouse gasses

97% of climate scientists agree that humans cause global warming (Cook et al., 2013). Many people have gotten used to driving or flying to holiday destinations and conferences, having electric appliances to make life easier, and enjoying an actively heated or cooled house. Typically fossil fuels are burned during these activities, causing emission of so called greenhouse gasses, which cause the earth to warm up. The United Nations Framework Convention on Climate Change has made an agreement, the so called Paris agreement (United Nations, 2016), to keep global warming below 2 ^{o}C.

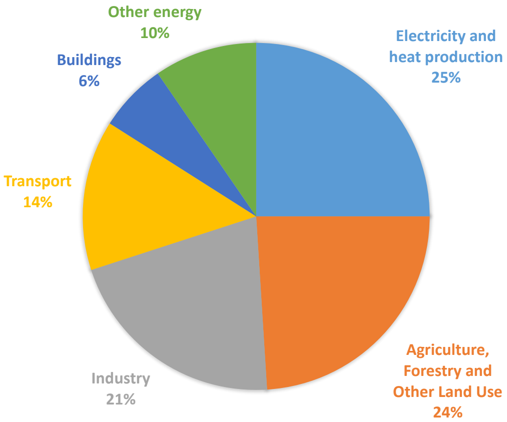


Figure 1.1: Contribution of greenhouse gasses per economic sector (Intergovernmental Panel on Climate Change, 2015).

Many people do not want to go back to pre-industrial era living standards, so the only way to comply with the Paris agreement is to make luxurious life more sustainable. The main activities that produce greenhouse gasses are depicted in Figure 1.1. Electricity and heat production are a combined total of 25% of greenhouse gas contribution (Intergovernmental Panel on Climate Change, 2015). Policies and visions are developed by governments to obtain concrete developments in specific domains such as transportation and the electricity grid (European Technology Platform SmartGrids, 2006). Engineers will have to play a major role in making life more sustainable by developing clever solutions that reduce greenhouse gas emissions. In the electricity grid, the traditional coal and natural gas fired power plants are being replaced by renewable energy sources such as wind mills and solar panels. These renewable energy sources do not emit greenhouse gasses during generation of electricity.

1.1.2 The electricity grid

Technological challenges appear due to the increasing penetration of renewable energy sources in the electricity grid. Instead of power being generated at a few dedicated locations, the generation becomes distributed over many different locations, e.g. solar panels at someone's home or wind mills scattered around the landscape. Power generated by wind mills and solar panels can be intermittent due to the weather, which makes it hard to continuously balance the supply and demand of electricity. Some of the issues that need to be addressed to successfully incorporate the renewables into the grid are the following (Olivares et al., 2014). Intermittent generation of power can be counteracted by appropriate levels of reserves, but economical operation under a reliability constraint has to be taken into account. Moreover the topology of the grid may change when participants go online or offline, so the market and control mechanisms need be able to handle this. The electricity grid has to undergo a transformation in order to be fully suitable for power generated by renewable sources.

A division into many microgrids is one of the ways to implement the smart electricity grid of the future (Hatziargyriou et al., 2007). A microgrid is formed from a local community of loads, renewable energy sources, and also energy storage and demand response services, as depicted in Figure 1.2. Each microgrid is connected to the main grid via a single interconnection, and can be operated either as connected to the grid, or as a standalone island. It is possible that a microgrid produces or consumes more than it can handle locally, and then this surplus or deficit needs to be transported to other parts of the grid, possibly across borders. When the microgrid is operating in grid connected mode, then it needs to contribute to the stability of the whole grid. This implies that communication between participants on a large scale plays a key role in the balancing of the grid of the future.

In the traditional grid configuration, power plants would be locally controlled to follow the targets set by a Transmission System Operator (Kundur et al., 1994). The way that supply and demand are balanced will need to change in a smart grid. A grid with distributed generation has so many generators that centralized control is no longer feasible. Participants in a microgrid are to coordinate among each other how much electricity is produced, and possibly how much is consumed. In this way the supply

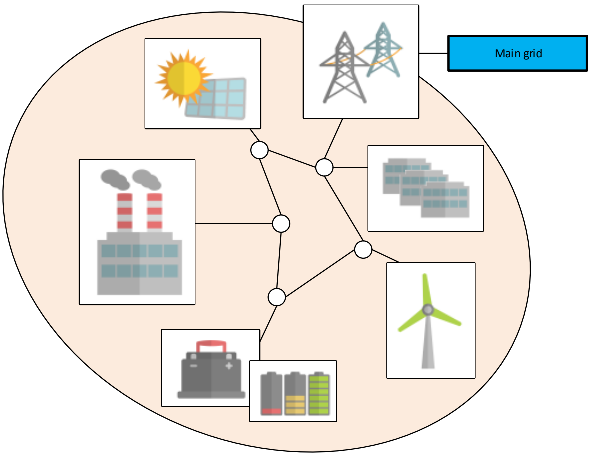


Figure 1.2: One microgrid where electricity is locally produced by windmills and solar panels, electricity is consumed by factories and houses, and where battery storage and a connection to the main grid provide a balance between supply and demand.

and demand is balanced locally as much as possible, with possibly some external help from the main grid.

Some of the challenges (Olivares et al., 2014) that need to be solved in these microgrids are described below. Local oscillations, instability or transient effects may plague the microgrid, and an analysis of the stability and effect of disturbances is necessary to guarantee smooth operation. Assumptions made for traditional power grid models may not be valid for microgrids, and the models need to be revised. A grid where renewables are prevalent may have low inertia, such that a small imbalance in supply and demand can lead to large deviations from the nominal frequency of 50 or 60Hz, which requires control mechanisms to react and prevent these imbalances. The systems and control field will need to address answers for the challenges in the microgrids.

1.1.3 Systems and control

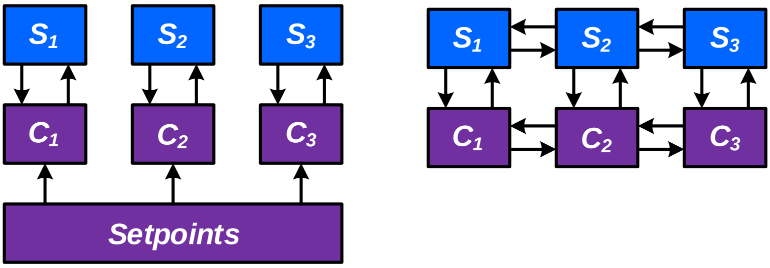
The field of systems and control specializes in the analysis of stability and oscillations, and the modeling and control of systems. The concept of a system must be defined, and for this we consider the following dictionary definition.

> **System:** *"A set of connected things or parts forming a complex whole, in particular: a set of things working together as parts of a mechanism or an interconnecting network ..."*
>
> —*The new Oxford dictionary of English, 1998.*

The tools for analysis and control design are traditionally tailored towards systems that operate in a single control-loop. A single system is controlled for a purpose, e.g.

to follow a setpoint or reject a disturbance. In case there are multiple controlled variables, the controller may work in a hierarchical structure where a central optimization algorithm generates setpoints for a number of separate control loops. In this control architecture each system is considered as an entity that does not interact with the others, as depicted in Figure 1.3a. Performance of the total system is improved by optimizing the performance of each local system with a local controller. This control strategy can lead to a globally optimal performance, but the weaknesses are that possible interactions between systems are neglected, and that it may be infeasible to synthesize a central governor when the number of sub-systems grows large.

In the more modern approach the sub-systems are seen as interacting entities, and controllers are designed to interact among each other, as depicted in Figure 1.3b. Considering multiple systems as an interconnected network of systems is quite natural, as this is included in the dictionary definition of a system. Here a controller communicates with a small number of other controllers in order to optimize performance. Interactions between systems are taken into account with this distributed control strategy, and controllers can be synthesized regardless of the size of the network.



(a) (b)

Figure 1.3: On the left: Sub-systems do not interact with each other, and each subsystem is individually controlled on the basis of setpoints generated on a higher level. On the right: Sub-systems and controllers interact among each other in order to satisfy a control objective.

1.2 Models

1.2.1 Definition

Many of the techniques applied in the systems and control domain depend on, or benefit from, the presence of a model. First it is established what is meant with a model by considering a dictionary definition.

> **Model:** *”A simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions.”*
> —*The new Oxford dictionary of English, 1998.*

According to (Eykhoff, 1974) a model is a simplification of reality that captures only the essential aspects. This implies that a model is a representation of the real world where irrelevant features are neglected, which makes it easier to understand the reality. There can be many forms of models, such as graphical, physical, verbal or mathematical. In a mathematical model this representation is a description in mathematical language, e.g. a set of equations.

A mathematical model may give us insight into a physical phenomenon. Newton's second law states that force applied to a mass is proportional to the acceleration of that mass, i.e.

$$F = ma,$$

where F is the force, m is the mass, and a is the acceleration. This model provides some understanding of how the world behaves, which is why models can help to improve the design of a system. *Dynamic* models are models where time plays a role in the described phenomenon, i.e. where a present value has an influence on a future value. An example is that the present-time acceleration of a mass has an influence on the future speed and location of the mass. If this data is available up to the current moment, then the model may be used to predict the future behavior of the modeled aspect. These predictions may be used to influence the future behavior of the system. Typical engineering systems can be represented by dynamic models, and their future behavior can be influenced with the application of a controller.

Design of a controller for a single control-loop can be improved with the use of models. In fact a controller can be designed such that it is the optimal controller for that model in some sense. Some controllers make explicit use of a model of the system. Also for distributed controllers a model of the system is an important tool needed to reach optimal performance. Moreover models of a distributed system can be used for stability analysis or to prevent oscillations. A list containing a number of possible applications of dynamic models is:

- While designing systems, a **simulation** using the model can reveal undesirable behavior, such that the design may be adapted to prevent this behavior.
- Models can be used to **estimate** an unmeasured quantity, for example the state of charge of a battery can be estimated on the basis of the measured voltage and current.
- **Forecasting** the future, for example as in a weather forecast, can be done on the basis of a dynamic model.
- **Faults** can be **detected** when the behavior of a system is no longer the same as the behavior of the model.
- A **controller** applied to a system can be **designed** to work optimally on a model of that system.
- The building of a model can provide new **insight** in how a system functions.

1.2.2 Dynamic network models

Engineers are designing systems that are larger and more advanced than before. Large-scale systems are composed as an interconnected network of smaller sub-systems. An example of such a large-scale system is the electricity grid. The general observation in the systems and control domain is that the considered systems also are growing in scale. Local sub-systems are locally controlled, and possibly the controllers will be communicating among each other. In order to decide which controllers are to communicate with which other controllers, it is important to know which sub-systems are interconnected to which other sub-systems, and how these interactions behave. This means that the interactions between sub-systems are an essential part of the behavior of the system. Then according to the definition of (Eykhoff, 1974) these interconnections and interactions should be captured in the model. The name *dynamic network model* will be adopted for models that capture the interconnection structure as well as the dynamics of a system.

A dynamic network will be constructed from two types of components and is separated from the unmodeled reality by a boundary:

- **Nodes** represent variables that depend on time, which possibly can be measured. Examples are the force and acceleration in Newton's second law.
- **Modules** represent the interconnections between nodes in a network, which contain the dynamic behavior. The mass plays this role in Newton's second law as it connects the force and acceleration nodes.
- The **boundary** represents the distinction between what part of reality is considered part of the network, and what is not. A dynamic network interacts with the world across the boundary, which causes a response in the dynamic network. As an example, someone may attempt to push a mass, generating a force that acts on the mass, which then makes the mass respond by moving.

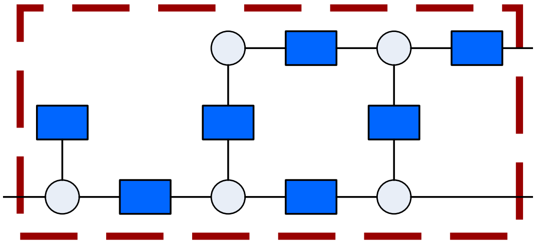


Figure 1.4: Graphical representation of a dynamic network. Nodes are represented by circles, modules are represented by rectangles. The boundary is depicted as the red dashed line.

An example of a graphical representation of a network is depicted in Figure 1.4. In the example it can be observed that nodes are connected with only a few other nodes, such that they interact directly only with a few other nodes. A dynamic network model has the property that the behavior of one node can be described fully by the neighboring nodes and the corresponding modules.

A microgrid can be used as an example for a dynamic network model. Variables such as power flows, deviation from the main grid frequency, or voltages can be used as nodes in the network. Modules represent the loads, generators and controllers that are connected to the grid. The structure of how the different modules are interconnected, and how the control layer is connected can be graphically represented as in Figure 1.5.

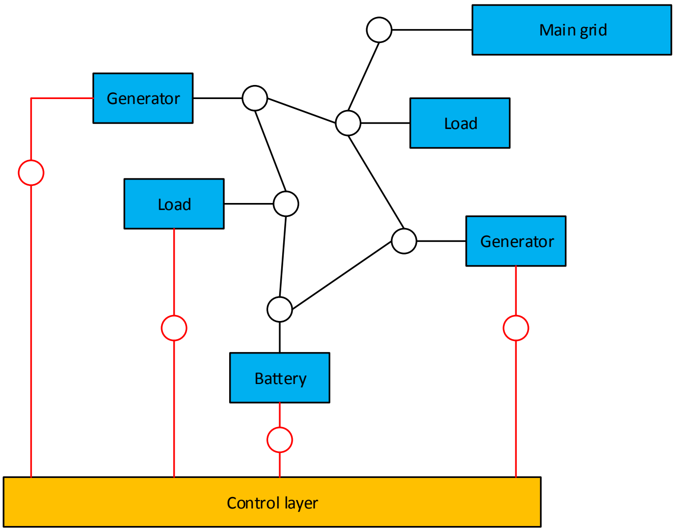


Figure 1.5: One microgrid and a control layer depicted as dynamic network model.

Next to the electricity grid, plenty of other engineering systems can be interpreted as a dynamic network. Each of the economic sectors described in Figure 1.1 has challenges where dynamic network models may be helpful. As an example, in the transportation sector the interactions between an internal combustion engine and an electric motor in the drive-train of a hybrid vehicle may be modeled as a dynamic network. This includes a model of the battery, for which it is important to estimate the State of Charge based on models (Beelen et al., 2016) to prevent damage and ensure optimal performance.

Outside of the engineering domain and unrelated to greenhouse gasses there are many possibilities to apply dynamic network models. The network model can play an important role every time a dynamic system model is needed in a situation where interconnections are a relevant feature. For example in systems biology, networks of interacting genes can be represented using a dynamic network model (Yuan et al., 2011) to provide insight into the nature of genes. Another example field is the stock market, where the interactions between different stocks can be modeled as a dynamic network (Materassi and Innocenti, 2010), which may be used to forecast the future values of these stocks.

Power grids and other fields of engineering and science raise a number of questions regarding dynamic network models. Among these questions are stability of the network and various control problems. Having an available network model can be helpful in solving these issues. What is often assumed is that a model is available, avoiding the question of how to obtain a model. When it is unknown how to model a network, then we have to be careful with the assumption that a model is available. Therefore, *the objective of this thesis is to contribute to the modeling of dynamic networks.*

1.3 Research question

The objective of modeling dynamic networks is quite broad and in this section the scope is narrowed down to a particular research question. Restrictions of the scope are made by discussing what types of network models are relevant, and what methods are available for obtaining these models. There are some different objectives for the modeling procedure, and these have their own specific challenges. Literature has a number of answers for the network modeling objective, but many open questions remain to be discussed.

1.3.1 First principles modeling vs. data-driven modeling

With the availability of cheap sensors it becomes possible to measure almost everything. Smart meters are being introduced in the grid such that the electricity consumption of clients can measured and sent automatically to the supplier. The availability of enormous amounts of data is both a blessing and a curse known as 'big data'. There is a tremendous amount of data available, but it is challenging to recover the relevant information from this data.

Typically there are two types of approaches to obtain mathematical models, first-principles modeling, and data-driven modeling. In first-principles modeling, known models from physics or other sciences are combined into a model of the desired system. With this approach it is required that the available equations are able to describe the behavior in a sufficiently accurate way, and that particular parameters of the equations can be determined with sufficient accuracy. This way of modeling can be a time consuming and therefore costly endeavor.

The alternative is a data-driven approach, which is based on the fact that measurements of a system contain information on the behavior of the system. In the classical reasoning of (Ljung, 1999) a user selects a *model set* containing candidate models with to be determined parameters, and then a *criterion* or *algorithm* selects a model from the set based on the available *measurement data*. An accurate model is selected by the criterion when the set of candidate models contains an accurate approximation of the real system, the data contains sufficient information, and the selection criterion is suitable. In this way it is possible to obtain accurate models, even when it is not known what kind of physical system has generated the data. All that is obtained is a model that mimics the behavior of the system, which is ultimately the relevant part of the model.

The difficulty with data-driven modeling is that measured data is used, which contains unknown disturbances. A criterion has to select an appropriate model, even though the models can not exactly explain the data due to disturbances. The model that is ultimately selected fits the data the best in some sense.

In this thesis the data-driven modeling approach of (Ljung, 1999) is followed. This implies that an appropriate set of candidate models must be chosen, and that an appropriate identification criterion must be chosen.

1.3.2 Linear time-invariant models

In reality the behavior of a system is usually of a non-linear nature, i.e. the principle of superposition does not hold between variables. Moreover this behavior often changes over time due to wear or changing conditions in the environment. When modeling on the basis of data the incorporation of non-linear or time-varying behavior involves many additional complexities compared to linear time-invariant (LTI) behavior. Additionally, in order to model non-linear behavior there are additional requirements on the measured data since all the complicated behavior must be excited if it is to be modeled. Theory for data-driven modeling of LTI dynamical systems is more advanced than theory for modeling non-linear or time-varying alternatives. Many real world systems can accurately be described by linear time-invariant approximations when operated around a working point. LTI networks are a good starting point, and so this thesis is restricted to the LTI setting. In the future the work may be extended to non-linear or time-varying situations.

1.3.3 Different model sets

The objective is to model network systems on the basis of data, and it has been established that a suitable set of models must be chosen. Networks of linear time-invariant systems can be described by for example ordinary differential equations or by discretized partial differential equations. A number of different LTI models suitable for dynamic network modeling exist in the literature and are considered.

Behavioral models

In the behavioral modeling paradigm (Willems and Polderman, 2013), a model describes simply which trajectories a signal is allowed to follow. Signals that are part of a behavioral model are not explicitly designated as either input or output. Nodes in a dynamic network model are also not explicitly designated as either input or output. However, when no structural restrictions are imposed to the behavioral model, then it does not capture the interconnection structure of modules in a dynamic network in a unique way. Moreover, unknown disturbances are not part of the behavioral modeling framework. For these reasons the behavioral models are not chosen as the modeling framework in this thesis.

Graph models

Graph models (Gross and Yellen, 2005; Bang-Jensen and Gutin, 2008) are models consisting of nodes and weighted links between those nodes. The graphical representation of a graph model makes it really clear which nodes are related to each other, and which nodes are the important ones. These models are widely used to represent a wide range of systems such as a network of roads, electrical circuits, or social networks. Dynamics can be modeled by including additional nodes that represent the time dependent behavior, however this can make a network needlessly complex. Dynamic networks can be considered as a special kind of graph model that includes dynamic behavior in the links, such that there are no additional nodes necessary. Properties and insight from graph theory can therefore often be transferred to the dynamical networks.

Probabilistic models

Unknown disturbances are often modeled as stochastic disturbances, which can be used to consider the nodes as stochastic variables. When nodes are described as stochastic variables, then conditional probabilities can describe relations between nodes in a structured way. A stochastic network can be defined where the stochastic variables form nodes of the network, and where conditional probabilities form the modules (Koller and Friedman, 2009). A definition of causality and the relation to probabilities is provided by (Granger, 1980). More modeling frameworks that are related are Structural Equation Modeling (Bollen, 1989), and the Structural Causal Model (Pearl, 2009).

State-space network models

It is well known that differential equations can be transformed into state-space models. The states can be considered as nodes in a graph model, and the relations between the states can be considered as the weighted links of that graph model. In (Massioni, 2014) distributed control of large-scale systems using state-space networks is discussed. Identification of state-space networks is also considered in literature, see for example (Haber and Verhaegen, 2014).

Transfer function network models

Modules in a network can be represented by transfer functions, such that the nodes are both the inputs as well as the output of the modules. A network of transfer functions can be related to the state-space model, and named Dynamic Structure Function (DSF) (Gonçalves and Warnick, 2008). Impulse responses, a particular type of tranfer function, are used by (Chiuso and Pillonetto, 2012) to describe a dynamic network. Wiener filters have a particular relation to transfer functions, and these are used by (Materassi and Salapaka, 2012) as modules. A setup of dynamical networks using transfer functions that includes models for unknown disturbances is introduced by (Van den Hof et al., 2013).

1.3.4 Identification methods and objectives

Some of the different network models described in the previous section are related with identification methods. These methods describe the criterion, or algorithm, that selects a model from the model set.

Subspace identification

State-space models are typically obtained by subspace identification methods, which are based on the realization of state-space systems from the Markov parameters (Ho and Kálmán, 1966; Van Overschee and De Moor, 2012). Dynamic network applications of subspace identification can be found for example in (Haber and Verhaegen, 2014), where network models of large-scale systems are identified. In the network version of subspace identification methods the interconnection structure has to be included into the algorithm.

Prediction error methods

Transfer function models are often obtained with the use of prediction error methods (Ljung, 1999), which under some conditions are equivalent to Maximum Likelihood estimators. An attractive property of Maximum Likelihood and prediction error methods is that asymptotically models with zero bias can be obtained with the minimum amount of variance. Prediction error methods have been extended to the dynamic network situation in (Van den Hof et al., 2013), although there are many open questions remaining.

Bayesian approach

Compared to Maximum Likelihood estimators, the Bayesian estimators can make use of additional a-priori information to improve an estimate. A Bayesian estimator may improve the Mean Squared Error of the model over the ML estimate if this a-priori information is available and accurate. A network model with semi-parametric impulse response models is used in (Chiuso and Pillonetto, 2012) for detection of the network topology using Bayesian identification methods.

1.3.5 What to identify?

The stated objective is to contribute to the modeling of dynamic networks, but what particular aspect do we want to model? Three main objectives of identification can be distinguished, and these match with different requirements on a-priori knowledge.

Full network identification

The objective to identify the full network is the most straightforward. All modules in the network are to be identified from data. This objective usually requires that the interconnection structure of the network is known, at least up to a sufficient degree.

The objective in (Gonçalves and Warnick, 2008) is to estimate the whole network from an estimated open-loop response, and in (Yuan et al., 2011; Yuan, 2012) unknown disturbances are added to the problem setting. The prediction error method is used to directly estimate the network in (Yue et al., 2017) without an intermediate step of estimating the open-loop response.

Single module identification

An alternative objective is to focus on identification of a single module in the network. Since the module can be fully described by nodes that are in the neighborhood of the target module, the required a-priori knowledge for this objective is reduced compared to full network identification. Only the topology around the single module has to be known up to a sufficient degree.

Estimation of a single module has been the objective in (Van den Hof et al., 2013) by extending the closed-loop prediction error methods known as direct method, two-stage method and joint-io method to the dynamic network. This reasoning was continued in (Dankers et al., 2015) for an instrumental variable method in an errors-in-variables situation. A prediction error method is combined with Bayesian estimation in (Everitt et al., 2018). When identifying a local module in a network it is not necessary to measure everywhere, which was shown by different approaches (Dankers et al., 2016; Linder and Enqvist, 2017a; Bazanella et al., 2017).

Topology identification

In situations where the interconnection structure of the network is unknown, the objective usually is to detect this interconnection structure, which is also known as topology detection. A typical way to perform this topology detection is by identifying the topology as well as all the network dynamics simultaneously, as described in the following papers.

Linear regression models are combined with the compressive sensing method in (Sanandaji et al., 2011, 2012) to decide which nodes are interconnected. Linear regression models are also used in (Chiuso and Pillonetto, 2012; Zorzi and Chiuso, 2017), but here topology is detected using Bayesian methods. This Bayesian approach has the advantage that sparseness of estimated topology does not depend on tuning by a user. In (Rojas and Hjalmarsson, 2011) a sparse topology is estimated by making use of Akaike's Information Criterion.

Alternative to identifying the exact topology one can focus on identification of topological properties. As an example, in (Mauroy and Hendrickx, 2017) the number of neighbors connected to a particular node is estimated.

Discussion

Identification of the full dynamic network, under the assumption that the topology is known, is seen as the base problem. Single module identification is an extension of full network identification in the sense that the conditions under which this is possible can be relaxed. Topology detection is another extension of full network identification in the sense that additional topology detection mechanisms have to be added to the estimation problem.

1.3.6 Open questions

A number of open questions and challenges remain in the network identification literature discussed so far. These aspects are discussed in this section, which then lead to the research question.

Experimental setup

In literature on network identification typically some experimental setup is assumed, and then some results are shown. With experimental setup is meant that assumptions are made on presence and correlation of unknown disturbances, presence of external excitations, and the allowed topologies or dynamics. Often different experimental setups are used, which can make it difficult to compare methods and results.

Restrictions on the unknown disturbances are common in the literature. Especially the assumptions that unknown disturbances are mutually uncorrelated, that they enter the network at every node, or that they are have a known spectral density are common. These assumptions are practical only when it is known from a-priori knowledge that they are satisfied. When this a-priori knowledge is not available, or when the assumptions are not applicable, then the restrictions need to be relaxed.

Disturbances in a network can be generated by many different unmeasured phenomena, and knowledge of the spectral contents of the noise is therefore unrealistic as an a-priori assumption. In system identification a number of solutions have been developed that may be employed in the dynamic network setting.

In an estimation problem, correlated disturbances and confounding variables can lead to biased estimates. Absence of confounding variables or uncorrelated disturbances is a typical assumption when models are obtained from data, but it is an open question whether this is necessary. In practice it seems plausible that disturbances may become correlated, for example as disturbance v_2 in Figure 1.6 that affects two nodes. We model confounding variables as a correlated disturbance, and choose to include correlated disturbances into the experimental setup, such that we may formulate ways to deal with the problem.

Another restriction that is common in literature, but that may not be practical, is the assumption that a noise must enter at every node. Modules in a network can be implemented controllers, and controller outputs can be noise-free, as e.g. typically considered in a classical closed-loop identification problem (Ljung, 1999). In this case

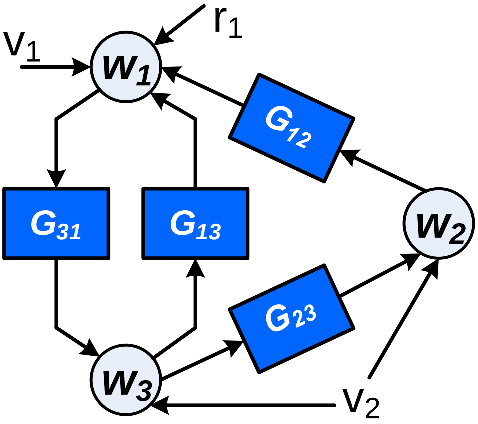


Figure 1.6: Example of a network with rank-reduced and correlated noise. Node signals are w_i, interconnected by modules G_{ij}, and perturbed by non-measured disturbance signals v_i. Signals r_i are excitation signals available to the user.

there is no process noise on a particular node signal. The more general case of noises with rank-reduced spectral density is also considered in order to represent situations where noise is dominated by a few sources. This latter situation is depicted in Figure 4.3 where the process noises on nodes 2 and 3 are the same (perfect correlation). When identifying the full network dynamics, the rank-reduced noise causes some fundamental issues that need to be addressed in the identification problem.

There exists another type of disturbance that does not affect the network itself, but that is present in data. This disturbance is caused by the sensors that measure nodes in a network, and is therefore known as *sensor noise*. In typical estimation problems sensor noise leads to difficult Errors-in-Variables problems (Söderström, 2018). Sensor noise may be handled in alternative ways, for example with instrumental variables (Dankers et al., 2015), but there are open questions remaining on this topic.

With the flexible experimental setup that is considered, it is a question which knowledge of this setup is available a-priori to the identification procedure. For example it may not be known whether some noises are correlated. It is important to determine which a-priori knowledge of the experimental setup is necessary for the identification of a network. When the necessary conditions are known, then it may be determined where to excite a dynamic network and where to measure, such that these conditions can be satisfied.

Variance

When unknown disturbances are modeled as stochastic processes, then estimated models become stochastic variables. A desired property for the estimated model is that the variability is low, i.e. that estimates based on different data sets differ by only a small amount. This variability is well described by the variance of the estimated model. For unbiased estimates the variance of an estimator is bounded from below by the Cramér-Rao Lower Bound (CRLB) (Rao, 1945). When the Cramer-Rao lower bound on the variance is achieved, then no other estimator can achieve an unbiased estimate with lower variance using the same model set and data set, i.e. the variability of the estimate is as low as possible. A classical estimator that has both the consistency property and variance at the CRLB is the Maximum Likelihood estimator (Rao, 1973).

For open-loop and closed-loop MIMO estimation it can be shown that the prediction error method can be equivalent to the maximum likelihood estimator (Ljung, 1999; Söderström and Stoica, 1989). For dynamic network identification the Maximum Likelihood analysis remains an open problem. A Maximum Likelihood estimator serves as a benchmark for other methods. Since the network identification methods (Van den Hof et al., 2013) are extensions of the closed-loop prediction error method, it is expected that Maximum Likelihood results can be obtained for network identification too.

The effect of some specific experimental setups on the variance of an estimated model has been investigated in (Everitt et al., 2013, 2015). However this analysis has not been performed for general network structures. Moreover general expressions for the variance of an estimated network model have not been obtained.

Algebraic loops

Dynamic networks can be defined in a way that nodes have an algebraic relation with each other, leading to the presence of algebraic loops. Applying the direct identification method in a closed-loop system, or a dynamic network, under presence of an algebraic loop does not lead to consistency (Ljung, 1999; Van den Hof et al., 2013). Other prediction error identification methods for dynamic network identification, e.g. the two-stage method and the instrumental variable method (Van den Hof et al., 2013; Dankers et al., 2015), are able to reach consistency by projecting node signals onto external variables under the presence of algebraic loops. When projecting onto external variables, the noise contributions are decorrelated. In a situation that process noises provide excitation, decorrelating the noise removes excitation and therefore information. When projecting onto external variables the variance is not minimized. Moreover the external variables are required to excite all dynamics in a sufficient way.

Algorithms

Network identification methods are developed to be applied in practice. However often the focus in prediction error methods is on the theoretical properties, without considering how these methods can be applied in practice in the form of an algorithm. Typically non-convex optimization problems are formulated that may run into local minima, which are sub-optimal models. For large-scale problems typical strategies to solve the prediction error method may fail or be prohibitively time consuming.

1.3.7 Main research question

A number of different choices can be made with regards to the identification objective, the network model, and the identification method. Estimation of the full network is seen as the base case, and for this case Maximum Likelihood estimates have not been obtained. A prime candidate to obtain Maximum Likelihood estimates are the prediction error methods, which are defined for transfer function network models. This then leads to the following research question.

> **Under which conditions can prediction error identification be applied, for efficient estimation of models of the full, or part of a, dynamic network?**

The next step is to formulate which particular parts of the research question to answer. Relevant literature is investigated to formulate sub-problems that have to be addressed.

1.4 Formulating sub-questions

The research question is essentially a summary of the topics that will be addressed in this thesis. There are a few key words in the research question that require some explanation.

- **Under which conditions:** The conditions relate to which assumptions on the experimental setup must be made in order to come to an identified model. This includes assumptions on the network topology, the correlation structure of unknown disturbances, and on the presence and location of external excitation.
- **Prediction error identification:** The scope of the research is limited to prediction error methods.
- **Efficiently:** This concept has a double meaning. On the one hand efficiency refers to statistical efficiency of estimates, meaning that all information in the data is explained by the model such that variance is minimized, like in a Maximum Likelihood estimate. On the other hand efficiency refers to algorithms that can compute the estimate rapidly and accurately for large-scale models.
- **Full, or part of a, dynamic network:** In this thesis two identification objectives are treated. Identification of all modules in a network, also known as identification of the full network, is one objective. The other identification objective is identification of a single module, which is identification of part of a network. The latter objective is treated as an extension of the full network identification problem.

All these topic will be addressed by formulating and answering a number of more detailed sub-questions. These sub-questions are introduced below.

1.4.1 Uniqueness of network models

With the dynamic network setup a flexible model for the data is chosen. It is possible that two network models of different topologies can represent the measured data equally well. This implies that not all modules can be identified from data, as the structural information contained in data is limited. We have to determine what information is present in data, and what information must be known a-priori from intuition or physical insight.

Part of the structural information can come from knowledge of the experimental setup. This insight can for example come from having defined the particular locations where reference excitations are injected. When knowledge of the experimental setup is available, only network models that match that knowledge are to be considered for identification, as the others can not be the data generating network. In this way we may avoid problems with non-uniqueness. The question to be answered is the following.

Under which conditions on the network model set can different network models, or modules, be distinguished from each other on the basis of measured data?

In literature the problem where different parameters can not be distinguished on the basis of data is known as the identifiability problem (Ljung, 1999). The situation where different network models can not be distinguished on the basis of data is a type of identifiability problem. Knowledge of the experimental setup that is incorporated into the model has an influence on the identifiability of networks.

Some assumptions are commonly made on the experimental setup, and these may influence whether the networks can be distinguished. It is common to impose restrictions on unknown disturbances, for example by assuming a known spectrum, or by restricting the allowed correlations. We will investigate whether these restrictions influence the uniqueness of network models, and consequently whether these restrictions are necessary. In the same way we investigate how much of the interconnection structure, or topology must be known. The presence and location of external excitation signals is another factor that influences the uniqueness of models. We investigate where these reference signals have to be injected.

As the main question we investigate whether the full dynamic networks can uniquely be determined from data. Uniqueness of the full network requires relatively many conditions to be satisfied, as every part of the model needs to be uniquely determined. A relaxation will be investigated where only uniqueness of particular parts of the network, e.g. modules, is required.

1.4.2 Full network identification

A central topic in this thesis is how to obtain a dynamic network model with minimum variance from a data set. Since different objective functions can lead to different properties of an estimate, what objective should we choose? Prediction error methods can be related to Maximum Likelihood estimates, which have the minimum variance property (Ljung, 1999). However these Maximum Likelihood properties have not been shown in literature for dynamic network estimators. Moreover the variance expressions of dynamic network estimators have also not been shown in literature.

In practical situations the disturbances of a network may be correlated or of a reduced rank. We are looking to extend the experimental setup to include flexible noise models. Correlated noises, confounding variables or rank-reduced noise leads to difficult identification problem that have not been resolved completely in literature. Estimating networks where noises may be correlated or rank-reduced is one of the objectives. Additional to estimating the networks, the objective is to obtain Maximum Likelihood estimates for the flexible dynamic network model. These reasons lead to the following question.

Can maximum likelihood estimates with minimum variance be obtained of a dynamic network for general noise conditions?

When Maximum Likelihood estimates are obtained for the general noise conditions, then the variance of the estimate should be at the lower bound. It is to be investigated what the variance expressions are, in particular for the rank-reduced noise situation. When these expressions are obtained, then it can be verified whether the variance is at the lower bound.

1.4.3 Algebraic loops

Having a flexible experimental setup allows for nodes to have an algebraic relation. This can occur in several situations, for example when a continuous time network is approximated with a discrete time network, or when the particular structure of a dynamic network originates from a structured physical system, where physical variables interact with each other, without the direct presence of a computer-controlled (digital) operation. An example of direct couplings between variables is force and acceleration in Newton's second law. Direct couplings between physical variables is then a natural situation to take into account.

Part of the question on the conditions is what information is present in data, and what information on the experimental setup has to be known a-priori. This part relates to the unique identification in the presence of algebraic loops. Both the presence and location of external signals as well as the required information contained in those signals are to be investigated.

Under which conditions can consistent estimates be obtained of a dynamic network that contains algebraic loops?

Since direct identification methods are not suitable for the situation of algebraic loops it is to be investigated what is the fundamental problem with estimating in presence of algebraic loops. We are looking for methods that get around the fundamental problems, which then lead to conditions under which we can identify the network. An additional question is then whether a method can be formulated that has reduced variance compared to the projection methods.

1.4.4 Local identification

When the objective is to estimate a local module in a network, then one approach is to identify the full network such that also the local module is obtained. This however is a conservative approach in the sense that all nodes must be measured and sufficiently excited. For the estimation of a single module it is not necessary to measure all modules, and one of the main issues in the single module identification problem is deciding which nodes are necessary to be measured.

Which selection of node signals allows for consistent estimation of a module of interest?

Multiple approaches can be followed to answer the question. The approach taken in this thesis is to consider the effect of removing certain nodes from a network model. In literature different ways of removing nodes from a network exist, and we are comparing and looking for ways to generalize these methods. Then we are looking for conditions under which it is possible to identify the module of interest from the newly obtained network.

1.4.5 Algorithm

Theoretical identification methods need to be accompanied by an algorithm to be applicable in practice. In prediction error methods there is usually no particular algorithm discussed, but these methods rely on optimization of a typically non-convex cost function. For network identification it is critical that the network topology can be encoded in the algorithm, which is not always possible in existing algorithms.

Additional issues with algorithms appear when considering networks of a large scale. The number of local minima in the non-convex cost function grows with the size of the network. It would be beneficial if the identification has effective algorithms to solve the identification, preferably in a way where local minima are avoided.

Which algorithms are suitable for the efficient identification of large-scale dynamic networks?

1.5 Overview of contents

1.5.1 Chapter 2

A precise definition of the dynamic network model is provided on the basis of a flexible experimental setup. Preliminary notions related to disturbances and network properties are also introduced. Then the basic concepts of prediction error methods are presented using a classical single-input-single-output system. Finally a state-of-the-art method for identification of dynamic networks is introduced.

1.5.2 Chapter 3

The question regarding uniqueness of the representation of dynamic networks is addressed. First the concept of network identifiability is defined and motivated, after which we are looking for conditions for network identifiability. Conditions on the unique representation of feedthrough terms and algebraic loops are addressed. Then checkable conditions on the experimental setup are presented to guarantee network identifiability. Identifiability of a single module instead of the network is shown to lead to less restrictive conditions. Then the conditions are shown to be checkable on the basis of the modeled network topology if we consider network identifiability in a generic sense.

This chapter is based on the following publications:

- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identifiability of linear dynamic networks. Automatica, 89:247-258, 2018,*
- *H.H.M.Weerts, P.M.J. Van den Hof, and A.G. Dankers. Single module identifiability in linear dynamic networks. arXiv preprint arXiv:1803.02586, 2018. (Accepted for presentation at CDC 2018),*

and indirectly on the preliminary work published in:

- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identifiability of dynamic networks with part of the nodes noise-free. In Proc. 12th IFAC Workshop ALCOSP, 2016. (IFAC-PapersOnLine, 49(13):19-24),* and
- *H.H.M. Weerts, A.G. Dankers, and P.M.J. Van den Hof. Identifiability in dynamic network identification. In Prepr. 17th IFAC Symposium on System Identification, 2015. (IFAC-PapersOnLine, 48(28):1409-1414).*

1.5.3 Chapter 4

The question regarding Maximum Likelihood estimation of all modules in a dynamic network is addressed by investigating the link with prediction error identification. A prediction error reasoning is built for dynamic networks that have correlated and rank-reduced noise, which is introduced as the joint-direct method. Then a Weighted Least Squares criterion is introduced that can be used to estimate the network. For the situation of minimum variance estimation in the presence of rank-reduced noise a Constrained Least Squares criterion is introduced. Maximum Likelihood estimates are then related to the estimates obtained by the two least squares criteria. It is then the objective to show that minimum variance estimates are obtained by the criteria under some conditions, and for this reason variance expressions are derived for the two least squares estimators.

This chapter is based on the following publication:

- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identification of linear dynamic networks with rank-reduced noise. Automatica, 98:256-268, December 2018,*

and indirectly on the preliminary work published in:

- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identification of dynamic networks with rank-reduced process noise. In Prepr. of the 20th IFAC World Congress. IFAC, 2017. (IFAC-PapersOnLine, 50(1):10562-10567),*
- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identifiability of dynamic networks with part of the nodes noise-free. In Proc. 12th IFAC Workshop ALCOSP, 2016. (IFAC-PapersOnLine, 49(13):19-24),* and
- *P.M.J. Van den Hof, H.H.M. Weerts, and A.G. Dankers. Prediction error identification with rank-reduced output noise. In Proc. 2017 American Control Conference, pages 382-387, Seattle, Florida, 2017.*

1.5.4 Chapter 5

The question on estimation of dynamic networks in the presence of algebraic loops is addressed here. Two predictor definitions and their implied predictor expressions are evaluated for the situation of having algebraic loops. The joint-direct method is formulated for one of the predictor definitions and a consistency analysis is performed. Some observations regarding required external excitations are made. The theoretical contributions are validated using simulation experiments.

This chapter is based on the following publication:

- *H.H.M. Weerts, P.M.J. Van den Hof, and A.G. Dankers. Identification of dynamic networks operating in the presence of algebraic loops. In proc. 55nd IEEE Conference on Decision and Control, pages 4606-4611, 2016.*

1.5.5 Chapter 6

The question of which nodes in a network need to be included in a model in order to be able to estimate a module of interest is addressed in this chapter. An answer is formulated by analyzing whether removing a node from a network leaves the module of interest invariant. The core mechanic that changes the dynamics of a module is the transformation of module dynamics that leave the node signals invariant. The notion of abstraction is introduced as a way to remove nodes by utilizing network transformations. Two existing abstraction methods that lead to network identifiable model sets are generalized. Then conditions under which the generalized abstraction algorithm leaves the module of interest invariant are investigated. A systematic way to choose a set of nodes such that the conditions for invariance are satisfied is shown, which then answers which nodes are to be measured.

This chapter is based on joint work with Jonas Linder. A paper on the basis of this chapter is in preparation for publication in a journal.

1.5.6 Chapter 7

The question regarding efficient algorithms that can perform identification in dynamic networks is addressed in this chapter. In order to formulate an algorithm a particular network ARMAX parameterization is defined for network models. A Sequential Least Squares algorithm on the basis of the network ARMAX model is defined. A consistency analysis of the introduced algorithm is made in the basis of equivalences with the Weighted NullSpace Fitting algorithm. Issues regarding implementation and practical usage are discussed, and some analysis is provided on the basis of simulations. Finally the performance of the introduced algorithm is compared to other algorithms on the basis of simulations.

This chapter is based on the following publication:

- *H.H.M. Weerts, M. Galrinho, G. Bottegal, H. Hjalmarsson, and P.M.J. Van den Hof. A sequential least squares algorithm for ARMAX dynamic network identification. In Prepr. 18th IFAC Symposium on System Identification, Stockholm, Sweden. IFAC, 2018.*

1.5.7 Other publications

Contributions have been made to publications that are not included in this thesis:

- *A.G. Dankers, P.M.J. Van den Hof, D. Materassi, and H.H.M. Weerts. Conditions for handling confounding variables in dynamic networks. In Prepr. of the 20th IFAC World Congress, 2017. (IFAC-PapersOnLine, 50(1):3983-3988),*

- *P.M.J. Van den Hof, A.G. Dankers, and H.H.M. Weerts. From closed-loop identification to dynamic networks: generalization of the direct method. In 56th IEEE Conference on Decision and Control, pages 5845-5850, 2017,* and

- *P.M.J. Van den Hof, A.G. Dankers, and H.H.M. Weerts. Identification in dynamic networks. Computers & Chemical Engineering, 109:23-29, 2018.*

2 The dynamic network & preliminaries

This thesis is about prediction error identification of dynamic networks. A classical system identification method is the prediction error method (Ljung, 1999; Söderström and Stoica, 1989), which has been used in many different ways for many situations. A definition of the dynamic network and its basic building blocks is provided. The noise model plays an important role in the thesis and is extensively elaborated upon. Then a review of the basic concepts of system identification using prediction error methods is provided. An extension to the state-of-the-art network prediction error identification methods is provided in order to formulate a solid basis for the remainder of the thesis. Finally the sub-questions posed in the introduction are related to the formally introduced concepts.

2.1 Network definition

In order to use prediction error methods in a *dynamic network*, a dynamic network model must first be defined. In this section a dynamic network model is formulated on the basis of the setup in (Van den Hof et al., 2013).

2.1.1 Basic building blocks

A dynamic network consists of L scalar *internal variables* or *nodes* w_j, $j \in \mathcal{N}$, where $\mathcal{N}$ is a set of cardinality L, and K *external variables* r_j, $j \in \mathcal{R}$, where $\mathcal{R}$ is a set of cardinality K. Each node is a basic building block of the network and is described as:

$$w_j(t) = \sum_{l \in \mathcal{N}_j} G_{jl}(q) w_l(t) + u_j(t) + v_j(t) \tag{2.1}$$

where q^{-1} is the delay operator, i.e. $q^{-1}w_j(t) = w_j(t-1)$;

- $\mathcal{N}_j$ is the set of indices of internal variables with direct causal connections to w_j, i.e. $l \in \mathcal{N}_j$ iff $G^0_{jl} \neq 0$.
- There are no self-loops in the network, i.e. nodes are not directly connected to themselves $j \notin \mathcal{N}_j$;
- G_{jl} are proper rational transfer functions that are referred to as *modules* in the network;
- u_j are generated by the *external variables* $r_k(t)$, $k \in \mathcal{R}$, via

$$u_j(t) = \sum_{k \in \mathcal{R}_j} R_{jk}(q) r_k(t), \tag{2.2}$$

  where $r_k(t)$ can directly be manipulated by the user and $\mathcal{R}_j$ is the set of indices of generating external variables with direct causal connections to w_j, i.e. $k \in \mathcal{R}^r_j$ iff $R^0_{jk} \neq 0$, and R_{jk} are proper rational transfer functions;
- v_j is *process noise.*

In a graphical way, each equation forms a building block as in Figure 2.1, where rectangles are used to represent the modules in the network.

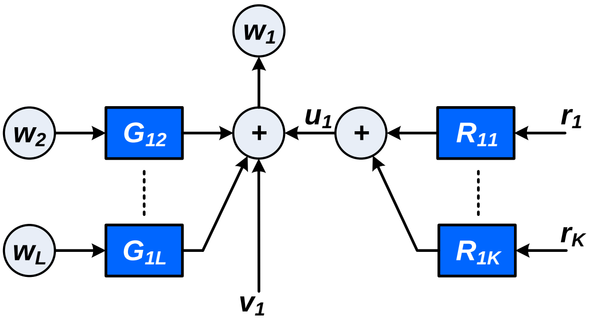


Figure 2.1: The graphical representation of (2.1) is a building block of the dynamic network.

Compared to the network setup in (Van den Hof et al., 2013), in (2.2) we allow multiple external variables to directly affect an internal variable, and an external variable may affect a node through a dynamic module. The additional decomposition is important for multiple reasons, it allows to specify where the external variables enter the network, and to represent the dynamic effect that an external variable can have on a node, and it gives the ability to represent an immersed network[1] within the same model class.

All the building blocks of the network are connected through the internal variables, such that the dynamic network is formed. We have not defined a particular way to

[1] The immersed network will be defined in Chapter 6

label the nodes, external variables and noises, but the common convention in writing will be to label nodes as $1, \dots, L$, and to label the external signals that enter at node j with label j. When combining the L node signals for the set of nodes $\mathcal{N} = 1, \dots, L$, the network expression is obtained

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1L} \\ G_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G_{L-1\ L} \\ G_{L1} & \cdots & G_{L\ L-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_L \end{bmatrix}. \tag{2.3}$$

The noise vector associated with the vector of internal variables $\left[w_1 \cdots w_L\right]^T$ is $v = [v_1 \cdots v_L]^T$, which is modeled as a stationary stochastic process with rational spectral density, such that there exists a p-dimensional white noise process $e := [e_1 \cdots e_p]^T$, $p \le L$, with covariance matrix $\Lambda > 0$ such that

$$v(t) = H(q)e(t).$$

The noise model will be specified in more detail in Section 2.1.3. In a straightforward matrix notation the dynamic network is represented as

$$w(t) = G(q)w(t) + R(q)r(t) + H(q)e(t). \tag{2.4}$$

The diagonal of $G(q)$ is 0 due to the absence of self-loops. The dynamic network can be graphically represented, for example as the example network in Figure 2.2.

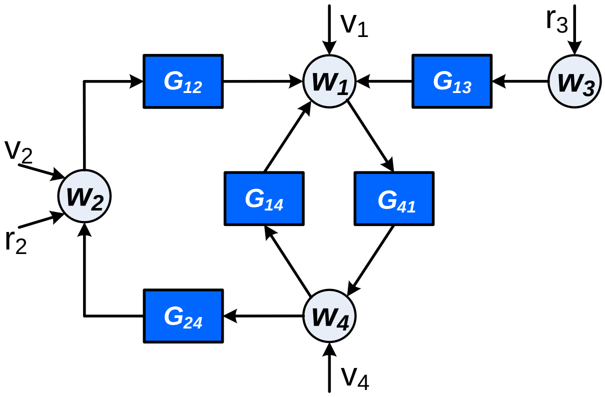


Figure 2.2: Graphical representation of a dynamic network, where circles are summation points with the indicated node as output.

Remark 2.1. *Elements in w are not necessarily ordered as $1, \dots, L$, and possibly the labels can be letters or other symbols, so element (i) of w does not necessarily correspond to w_i. The same holds true for element (i, j) of G, which does not necessarily corresponds with G_{ij}. In case element (i, j) of G is explicitly used, as opposed to G_{ij} then this will be denoted with $G_{(i,j)}$, and similar for w.*

2.1.2 Spectral density

In order to evaluate the spectral contents of signals a definition of the power spectral density is required. The cross power spectral density of vector signals $a(t)$ and $b(t)$ is defined as

$$\Phi_{ab}(\omega) := \mathcal{F}\{\mathbb{E}[a(t)b^T(t-\tau)]\}, \tag{2.5}$$

where $\mathcal{F}$ is the discrete-time Fourier transform, and $\mathbb{E}$ the expected value operator. The auto power spectral density of signal $a(t)$ is defined as

$$\Phi_{a}(\omega) := \mathcal{F}\{\mathbb{E}[a(t)a^T(t-\tau)]\}. \tag{2.6}$$

2.1.3 The noise model

Unknown disturbances may induce correlations between node signals that are not caused by the modules of the network. A noise model is made in an identification setting in order to explain these correlations.

The noise model $v(t) = H(q)e(t)$ requires some further specification. Associated with v is the spectrum $\Phi_v(\omega)$. In this thesis we consider the situation that the spectrum $\Phi_v(\omega)$ may not be full rank, and this situation is worked out in this section. Using spectral factorization two different representations of the noise model are derived, after which an implication for the probability density function is shown.

For $p = L$, referred to as the full-rank noise case, H is square, stable, monic and minimum-phase. The situation $p < L$ will be referred to as the *singular* or *rank-reduced* noise case. In this latter situation for notational simplicity and without loss of generality the following assumption will be made.

> **Assumption 2.2.** *The vector of node signals w is ordered in such a way that $\begin{bmatrix} v_{(1)} \cdots v_{(p)} \end{bmatrix}^T$ is a full rank noise process.*

When given just a set of measurements which contain rank-reduced process noise, it is not known what the rank and ordering of the noise process is. We have made the Assumption 2.2 that noise is ordered such that the first p nodes contain a full rank noise process. This assumption may look restrictive, and a question is whether the necessary ordering information and rank p are present in the data set. The question whether the ordering information is present in the data is an identifiability related question. If this information is present, it can be identified, and Assumption 2.2 would not be restrictive. A discussion on identification without a-priori knowledge of the ordering defined in Assumption 2.2 is made in Section 3.8.

To describe properties of H we need the following lemma, which is an adapted version of the spectral factorization theorem (Youla, 1961) that is also used in Weerts et al. (2018c).

Lemma 2.3 (Factorization of reduced-rank spectra). *Consider an L-dimensional stationary stochastic process x with rational spectral density Φ_x and rank $p < L$, that satisfies the ordering property of Assumption 2.2. Then*

a. *Φ_x allows a unique spectral factorization*

$$\Phi_x = F\Delta F^*$$

with $F \in \mathbb{R}^{L\times p}(z)$, $F = \begin{bmatrix} F_a \\ F_b \end{bmatrix}$ with F_a square, monic, and F stable and having a stable left inverse $F^\dagger$ that satisfies $F^\dagger F = I_p$, and $\Delta \in \mathbb{R}^{p\times p}$, $\Delta > 0$;

b. *Based on the unique decomposition of Φ_x in (a.), there exists a unique factorization of Φ_x in the structure:*

$$\Phi_x = \breve{F}\breve{\Delta}\breve{F}^*$$

with $\breve{F} \in \mathbb{R}^{L\times L}(z)$ monic, stable with a stable inverse and $\breve{\Delta} \in \mathbb{R}^{L\times L}$, having the particular structure

$$\breve{F} = \begin{bmatrix} F_a & 0 \\ F_b - \Gamma & I \end{bmatrix}, \qquad \breve{\Delta} = \begin{bmatrix} I \\ \Gamma \end{bmatrix} \Delta \begin{bmatrix} I \\ \Gamma \end{bmatrix}^T$$

and $\Gamma := \lim_{z\to\infty} F_b(z)$.

Proof. Part (a) is the standard spectral factorization theorem, see Youla (1961). The decomposition in part (b) can be verified by direct computation. Stability of $\breve{F}$ follows from stability of F. Stability of

$$\breve{F}^{-1} = \begin{bmatrix} F_a^{-1} & 0 \\ -(F_b - \Gamma)F_a^{-1} & I \end{bmatrix} \tag{2.7}$$

follows since it contains only stable components. ■

Lemma 2.3a shows that the noise process $v(t)$ can be represented as a unique factorization with an H that satisfies

$$H(q) = \begin{bmatrix} H_a(q) \\ H_b(q) \end{bmatrix}. \tag{2.8}$$

The feedthrough term of H_b will be indicated with Γ, i.e. $\Gamma := \lim_{z\to\infty} H_b(z)$. When we apply Lemma 2.3b to $v(t)$ we can make a unique factorization using the same H_a and H_b as in (2.8)

$$v(t) = \breve{H}(q)\breve{e}(t) = \begin{bmatrix} H_a(q) & 0 \\ H_b(q) - \Gamma & I \end{bmatrix} \begin{bmatrix} e \\ \Gamma e \end{bmatrix}, \tag{2.9}$$

where

- H is stable and has a stable left inverse $H^\dagger$, satisfying $H^\dagger H = I_p$, the $p \times p$ identity matrix;
- H_a is a proper rational transfer function which is square, monic, stable and stably invertible;
- and H_b is a stable proper rational transfer function which satisfies $(H_b - \Gamma)H_a^{-1}$ is stable.

The L-dimensional white noise process $\breve{e}$ has covariance matrix $\breve{\Lambda}$ defined by

$$\breve{\Lambda} = \begin{bmatrix} I \\ \Gamma \end{bmatrix} \Lambda \begin{bmatrix} I \\ \Gamma \end{bmatrix}^T . \tag{2.10}$$

From the definition of $\breve{e}$ we can see that there is a particular relation between the driving white noise process in the first p nodes and the last $L - p$ nodes.

With (2.8) and (2.9) there are actually at least two different noise model representations:

$$v(t) = H(q)e(t) = \breve{H}(q)\breve{e}(t).$$

In the case of full-rank noise, $p = L$, and both representations are the same. Both expressions will be utilized.

The white noise process $e(t)$ is modeled as a stationary stochastic process. The probability density function (pdf) of the rank-reduced process $\breve{e}$ is defined by two equations (Rao, 1973), i.e. the pdf of e and the additional constraint

$$\begin{bmatrix} \Gamma^0 & -I \end{bmatrix} \breve{e} = 0. \tag{2.11}$$

An interpretation of this characterization of $\breve{e}$ is a p-dimensional pdf that lives on a plane described by (2.11). This interpretation is illustrated in Figure 2.3 for an example of a 2-dimensional noise process $\breve{e}(t)$ having rank 1 with a Gaussian pdf.

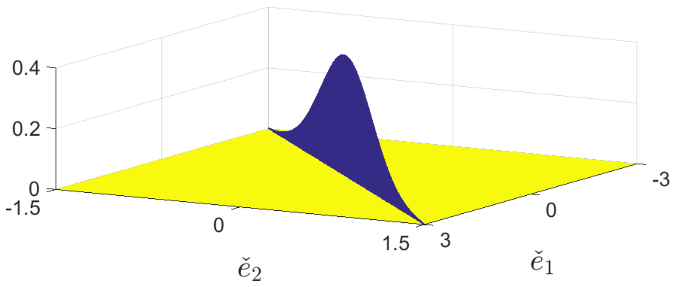


Figure 2.3: pdf of rank-reduced noise $\breve{e}(t) = [e(t)\ 0.5e(t)]^T$, with $e(t) \sim \mathcal{N}(0, \Lambda^0)$, a 1-dimensional random variable.

2.1.4 Sensor noise

Noise $v(t)$ is a process noise, which represents external unmeasured disturbances that affect the node signals. However there can be another source of noise from the sensors that measure the nodes. This sensor noise is affecting only the data, but has no influence on the actual network.

As an example of the difference between process and sensor noise, consider the measurement of temperature in a room. When the temperature changes due to an unmeasured disturbance, e.g. someone opens the window, then this is a process noise. When the measurement is disturbed while the temperature is unaffected, e.g. electromagnetic interference on the sensor, then this is sensor noise.

Node measurements affected by sensor noise are defined as

$$\tilde{w}(t) = w(t) + s(t), \tag{2.12}$$

where s is a stochastic process. The node measurements $\tilde{w}$ are then modeled by (2.12) combined with (2.4). In classical identification settings sensor noise would lead to an Errors-In-Variables problem, which is difficult to solve (Söderström, 2018).

It will now be shown that sensor noise can be written as a process noise through manipulation of the network equations. When substituting $w = \tilde{w} - s$ into the network equation (2.4),we obtain that $\tilde{w} - s = G(\tilde{w} - s) + v + Rr$, which can be manipulated to

$$\tilde{w} = G\tilde{w} + Rr + \underbrace{v + (I - G)s}_{:=\tilde{v}}. \tag{2.13}$$

With this substitution a new network representation is obtained with a modified process noise $\tilde{v}$, but with nodes $\tilde{w}$ instead of w. Even when the original noises v and s have a diagonal spectrum, then $\tilde{v}$ is unlikely to have diagonal spectrum due to the correlations induced by the G in $\tilde{v} = v + (I - G)s$.

The implication of the new process noise $\tilde{v}$ is that sensor noise in a dynamic network can be represented as a process noise that has a non-diagonal spectrum. In this thesis sensor noises are modeled as correlated process noises, and no particular attention is given to the sensor noise situation.

2.1.5 Open-loop response and spectra

In order to formulate what information is present in the data, the open-loop response of the network and the spectral density of the data must be defined. The network transfer function that maps the external signals r and e into the node signals w is denoted by:

$$T(q) = \begin{bmatrix} T_{wr}(q) & T_{we}(q) \end{bmatrix}, \tag{2.14}$$

with

$$T_{wr}(q) := (I - G(q))^{-1} R(q), \text{ and} \tag{2.15}$$

$$T_{we}(q) := (I - G(q))^{-1} H(q), \tag{2.16}$$

where the notation $T(q) = (I - G(q))^{-1} U(q)$ with

$$U(q) = \begin{bmatrix} H(q) & R(q) \end{bmatrix} \tag{2.17}$$

is also sometimes used. This is also known as the open-loop response of the network corresponding with

$$w(t) = T_{wr} r(t) + \bar{v}(t), \tag{2.18}$$

where noise component $\bar{v}(t)$ is defined by $\bar{v}(t) := T_{we}(q)e(t)$.

Nodes $w(t)$ can be represented as a spectral density in the following way

$$\Phi_w(\omega) = T_{wr}(e^{i\omega})\Phi_r(\omega)T_{wr}^T(e^{-i\omega}) + \Phi_{\bar{v}}(\omega). \tag{2.19}$$

Another useful spectrum is the cross power spectral density of w and r

$$\Phi_{wr}(\omega) = T_{wr}(e^{i\omega})\Phi_r(\omega). \tag{2.20}$$

The power spectral density of $\bar{v}$ is

$$\Phi_{\bar{v}}(\omega) := T_{we}(e^{i\omega})\Lambda T_{we}^T(e^{-i\omega}), \tag{2.21}$$

which can be determined from the known spectra using

$$\Phi_{\bar{v}}(\omega) = \Phi_w(\omega) - T_{wr}(e^{i\omega})\Phi_r(\omega)T_{wr}^T(e^{-i\omega}). \tag{2.22}$$

2.1.6 Paths and loops

Some notions from graph theory will be borrowed for use in the dynamic network. Modules form the interconnections between nodes. A node w_k is said to be an *in-neighbor* of node w_j if $G_{jk} \neq 0$, and w_j is said to be the *out-neighbor* of node w_k. As an example, in Figure 2.2 w_1 is an out-neighbor of w_3. Moreover w_1 is both the in-neighbor and out-neighbor of w_4.

Another notion that is useful is a *path* in the network, which is essentially a sequence of modules. More precisely there exists a path through nodes $w_{n_1}, \dots, w_{n_k}$ if

$$G_{n_1 n_2} G_{n_2 n_3} \cdots G_{n_{(k-1)} n_k} \neq 0.$$

A *loop* is a path that ends where it begins, more precisely a loop is a path where $n_1 = n_k$. As an example, in Figure 2.2 there is a path $w_4 \to w_1 \to w_2 \to w_4$ since $G_{41}G_{12}G_{24} \neq 0$ and this is also a loop. There is no path $w_3 \to w_1 \to w_4$ since $G_{31} = 0$.

2.1.7 Properness of modules

A transfer function G_{ij} has a *direct term* or *feedthrough* defined by

$$G_{ij}^{\infty} := \lim_{z\to\infty} G_{ij}(z).$$

The feedthrough of G_{ij} is non-zero when both G_{ij} and G_{ij}^{-1} are proper, and then G_{ij} is also said to be *bi-proper*. In terms of notation, for any transfer function $A(z)$ we will denote $A^{\infty} := \lim_{z\to\infty} A(z)$. Therefore $G^{\infty} := \lim_{z\to\infty} G(z)$ to indicate the direct terms of the whole network. A transfer function G_{ij} is said to have a delay when it is strictly proper, i.e. $G_{ij}^{\infty} = 0$.

Using these direct terms a special kind of loop can be defined: When all modules that are part of a loop have a non-zero feedthrough, then the loop is an *algebraic loop*, i.e. the nodes $w_{n_1}, \dots, w_{n_k}, w_{n_1}$ form an algebraic loop if

$$G^{\infty}_{n_1 n_2} G^{\infty}_{n_2 n_3} \cdots G^{\infty}_{n_k n_1} \neq 0.$$

When a permutation matrix Π can transform $\Pi G^{\infty} \Pi^T$ to a strictly upper triangular matrix, then there are no algebraic loops in the network (Deo, 1974).

2.1.8 Well-posedness

With the equations defined so far, a network model could be defined which can not appear in reality. For example the algebraic loop described by $G = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ would lead to a singular and non-invertible $(I - G)$ so that there is no open-loop response $T = (I-G)^{-1}[R \;\; H]$. Therefore it is required that the network is *well-posed* (Willems, 1971; Dankers, 2014), which is more precisely defined by the following definition.

> **Definition 2.4** (Definition 2.10 from (Dankers, 2014)). *Consider a dynamic network as defined in* (2.4)*. Consider a loop embedded in the network. Suppose the loop passes through the internal variables $w_{l_1}, \dots, w_{l_n}$. Let $\mathcal{L} = \{w_{l_1}, \dots, w_{l_n}\}$. If a variable in the network has a direct connection to any $w_l \in \mathcal{L}$ denote this variable as a 'variable that affects the loop $\mathcal{L}$'. Let $\mathcal{A}_{\mathcal{L}}$ denote the set of variables that affect the loop. Note that $\mathcal{A}_{\mathcal{L}}$ consists of internal variables, external variables, and process noise variables. The loop defined by $\mathcal{L}$ is well-posed if the following conditions are satsified:*
>
> *(a) The internal variables of the loop (i.e. all $w_l \in \mathcal{L}$) are completely (uniquely) determined by the variables that affect the loop, i.e. all variables in $\mathcal{A}_{\mathcal{L}}$.*
>
> *(b) The internal variables of the loop depend causally on the variables that affect the loop.*
>
> *(c) The internal variables of the loop depend on the variables that affect the loop in a continuous manner.*
>
> *(d) Small changes in the model should not result in a loop $\mathcal{L}$ that violates Conditions (a) - (c).*

For a more detailed explanation of the well-posedness property see (Dankers, 2014). Any dynamic network that corresponds to a physical system is well-posed, however some models that we define on paper do not make physical sense. The models that do not make physical sense are not well-posed, so requiring that a network model is well-posed is not a restriction on applicability. It is rather straightforward to check well-posedness of a network.

Proposition 2.5 (Proposition 2.14 in (Dankers, 2014))**.** *A dynamic network model as defined in* (2.4) *is well-posed if all principal minors of* $(I - G(\infty))^{-1}$ *are nonzero.*

As shown in the beginning of this section, an algebraic loop can lead to a non-well-posed model, however this is only true for specific cases. In general well-posed networks can have algebraic loops as part of the network. All networks which occur in reality are well-posed, so any network that is excluded due to this property can not exist in reality (Willems, 1971).

2.1.9 Dynamic network model

For the remainder of the thesis it is important to clearly define the network model. This definition is more general compared to network models defined in literature, as this model includes a flexible noise model and the possibility of algebraic loops.

Definition 2.6 (dynamic network model)**.** *A network model of a network with* L *nodes, and* K *external excitation signals, with a noise process of rank* $p \leq L$ *is defined by the quadruple:*

$$M = (G, R, H, \Lambda)$$

with

- $G \in \mathbb{R}^{L\times L}(z)$, *diagonal entries 0, all modules proper and stable*[2]*;*
- $R \in \mathbb{R}^{L\times K}(z)$, *proper;*
- $H \in \mathbb{R}^{L\times p}(z)$, *stable, with a stable left inverse, and satisfying the decomposition* $H(q) = \begin{bmatrix} H_a(q) \\ H_b(q) \end{bmatrix}$, *where* H_a *is stable, monic, stably invertible, and* H_b *is stable, proper, and satisfies* $(H_b - \Gamma)H_a^{-1}$ *is stable, where* $\Gamma = \lim_{z\to\infty} H_b(z)$.
- $\Lambda \in \mathbb{R}^{p\times p}$, $\Lambda > 0$*;*
- *the network is well-posed.*

Noise covariance matrix Λ is included in the definition of a model, as is common for multivariable models (Söderström and Stoica, 1989).

In this thesis dynamic network models are defined for the situation that all modules G are stable, while prediction error methods are in closed-loop systems able to identify

[2] The assumption of having all modules stable is made in order to guarantee that T_{we} (2.16) is a stable and stably invertible spectral factor of the noise process that affects the node variables.

unstable modules when the loop is stable (Ljung, 1999). Further research is required to make the theory presented in this thesis suitable for unstable G. One paper where unstable G are treated is (Bottegal et al., 2018).

The original network is unknown, and some identification criterion chooses the most suitable model to represent the network from a set on the basis by minimizing a cost function. A set of candidate models is defined from which the final model is chosen, and in this case the set of candidate models is a set of network models. Network models are parameterized by some parameter $\theta \in \Theta$, and the set of candidate network models is then defined on the basis of the parameter space Θ.

> **Definition 2.7** (network model set). *A network model set for a network of L nodes, K external excitation signals, and a noise process of rank $p \leq L$, is defined as a set of parametrized matrix-valued functions:*
>
> $$\mathcal{M} := \left\{M(\theta) = \big(G(q,\theta), R(q,\theta), H(q,\theta), \Lambda(\theta)\big), \theta \in \Theta\right\},$$
>
> *with all models $M(\theta)$ satisfying the properties as listed in Definition 2.6.*

Only model sets for which all models in the set share the same rank, i.e. $rank(\Lambda(\theta)) = p$, are considered in this thesis. Parameter θ is used only as a vehicle for creating a set of models. Particular properties of the mapping from parameters to network models are not considered.

2.2 Identification with prediction error methods

2.2.1 The basics

Here the three concepts that form the core of any system identification method are described. This section is based on the contents of (Ljung, 1999; Söderström and Stoica, 1989). First the basic concepts of system identification are discussed for a basic open-loop identification problem, after which prediction error methods are discussed for the closed-loop situation.

A system identification method describes the way to build a mathematical model of a system on the basis of measured data, such that the estimated model describes the data well. In the prediction error method a model is selected from a set of candidate models by some criterion. The core concepts are

- a data set,
- a model set, and
- an identification criterion.

After a model has been estimated from a data set, the model may be validated on the basis of validation data. Different validation methods exist, but this topic is not treated in this thesis.

The data set is the measured data that is generated by the data generating system $\mathcal{S}$, which is the system that is to be modeled. For single-input-single-output systems the data generating system is typically described by

$$y(t) = G^0(q)u(t) + v(t), \tag{2.23}$$

where u is known as the input of the plant G^0, y is known as the output, v is the unknown noise disturbance, t is the time index, and q^{-1} is the delay operator, i.e. $q^{-1}w_2(t) = w_2(t-1)$. The plant can be described by a rational transfer function

$$G^0(q^{-1}) = \frac{b_0 + b_1q^{-1} + b_2q^{-2} + b_3q^{-3} + \cdots}{1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3} + \cdots}. \tag{2.24}$$

In an experiment some signal $u(t)$ is designed and applied as an input to the plant, for example a multisine, and the output $y(t)$ is measured for the duration of the experiment. The data set contains $u(1), \ldots, u(N)$ and $y(1), \ldots, y(N)$, where N is the number of samples that were drawn. The unknown noise $v(t)$ is assumed to be a stationary stochastic variable, and $u(t), y(t)$ are assumed to be quasi stationary.

The model set is constructed by parameterizing the transfer functions associated with the model, i.e.

$$G(q,\theta) = \frac{\theta_{b_0} + \theta_{b_1}q^{-1} + \theta_{b_2}q^{-2} + \cdots + \theta_{b_n}q^{-n}}{1 + \theta_{a_1}q^{-1} + \theta_{a_2}q^{-2} + \cdots + \theta_{a_n}q^{-n}}, \tag{2.25}$$

where $\theta = \begin{bmatrix}\theta_{a_1} & \theta_{a_2} & \cdots & \theta_{a_n} & \theta_{b_0} & \theta_{b_1} & \theta_{b_2} & \cdots & \theta_{b_n}\end{bmatrix}^T \in \Theta$. Set Θ is the space that contains the parameters of the parameterized model, and so it governs the model set. The *model set* $\mathcal{M}$ is the set of all models that have $\theta \in \Theta$.

An identified model is then obtained from the identification criterion, which is a function that selects a model from the set of candidate models, such that the chosen model describes the data set well in some sense. Therefore the criterion is a function of $u(1), \ldots, u(N)$ and $y(1), \ldots, y(N)$, and the model set. A criterion based on the error described by $(G^0 - G(\theta))$ seems logical, however G^0 is an unknown quantity. For that reason the data is used, as the data contains the necessary information on G^0. A good model is able to describe the output well on the basis of the input, so an error that can be optimized is then

$$\varepsilon(t,\theta) = y(t) - G(q,\theta)u(t), \tag{2.26}$$

where $G(q,\theta)u(t)$ is essentially mimicing the output. A criterion function that is often used is the least squares criterion

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t,\theta), \tag{2.27}$$

where the parameters θ is chosen to minimize the power of error $\varepsilon(t,\theta)$.

2.2.2 Prediction error methods

Some network identification methods in literature, and the identification methods treated in this thesis are based on the prediction error method (Ljung, 1999). In the prediction error method a thorough reasoning is added to the choice of the error and the criterion by using a *predictor*. Under the assumption that G^0 is causal, the one-step-ahead predictor is defined as

$$\hat{y}(t|t-1) := \mathbb{E}\Big(y(t)\Big|y(t-1)^-, u(t-1)^-\Big), \tag{2.28}$$

where $y(t-1)^-$ is a shorthand notation for $y(t-1), y(t-2), \cdots$. This predictor is able to predict the value of $y(t)$ using only past and present values of u and y, so it is an excellent way of representing dynamic relations in data. The predictor will be used to explain the dynamics in the data in an identification setting.

When the noise $v(t)$ is non-white, i.e. $v(t)$ is correlated with $v(t-1)^-$, then to evaluate the predictor expression the noise must be modeled, which can be done by writing the noise as filtered white noise

$$v(t) = H^0(q)e(t), \tag{2.29}$$

where H^0 is a monic rational transfer function and $e(t)$ is a white noise with bounded moments. With this noise model the predictor (2.28) results in the expression (Ljung, 1999)

$$\hat{y}(t|t-1) = (H^0)^{-1}G^0(q)u(t) + \left(1-(H^0)^{-1}\right)y(t). \tag{2.30}$$

The parameterized model additionally includes a noise model

$$H(q,\theta) = \frac{1+\theta_{c_1}q^{-1}+\theta_{c_2}q^{-2}+\cdots+\theta_{c_n}q^{-n}}{1+\theta_{d_1}q^{-1}+\theta_{d_2}q^{-2}+\cdots+\theta_{d_n}q^{-n}}. \tag{2.31}$$

This is used to define the parameterized predictor

$$\hat{y}(t|t-1;\theta) = (H(q,\theta))^{-1}G(q,\theta)u(t) + \left(1-(H(q,\theta))^{-1}\right)y(t), \tag{2.32}$$

and the *prediction error*

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|t-1;\theta) = (H(q,\theta))^{-1}\Big(y(t) - G(q,\theta)u(t)\Big). \tag{2.33}$$

In the prediction error method, this prediction error is optimized in criterion (2.27).

2.2.3 Concepts in prediction error methods

Analysis of the estimated model is an important part of the prediction error method. It is important for an estimate to be an accurate representation of the real system in some sense. A way to define accuracy is *consistency*, which describes that an estimated model tends to the data generating model as more data is collected. For consistency it is required that the model $G(q,\theta)$ is flexible enough to be an accurate description of $G^0(q)$, i.e.

$$\exists\theta_0 \in \Theta \text{ s.t. } G(q,\theta_0) = G^0(q). \tag{2.34}$$

In (Ljung, 1999) an asymptotic analysis is presented that guarantees consistency of an estimate, i.e.

$$\hat{\theta}_N \to \theta_0 \text{ w.p. } 1 \text{ as } N \to \infty, \tag{2.35}$$

under some conditions, which has the implication that

$$G(e^{j\omega}, \hat{\theta}_N) \to G^0(e^{j\omega}). \tag{2.36}$$

In order to formulate the conditions for consistency two concepts need to be introduced, namely *identifiability* and *informativity*, and this is done next.

The notion of identifiability is a classical notion in system identification, and the concept has been used in different settings, so it becomes important to specify what is meant with network identifiability. The classical identifiability definition as present in (Ljung, 1976; Söderström et al., 1976) is a consistency-oriented concept concerned with estimates converging to the true underlying system (system identifiability) or to the true underlying parameters (parameter identifiability). In current literature, identifiability is defined as a property of a parametrized model set, referring to a unique one-to-one relationship between parameters and predictor model, see e.g. (Ljung, 1999). As a result a clear distinction has been made between aspects of data informativity, a property of measured data, and identifiability. For an interesting account of these concepts see also the more recent work (Bazanella et al., 2010).

A necessary condition for consistency is that only one point in the parameter space $\theta \in \Theta$ is associated with the data generating system. This uniqueness concept is defined in the following way.

Definition 2.8 (Identifiability (Ljung, 1999)). *Model set $\mathcal{M}$ is globally identifiable at $\theta_1 \in \Theta$ if for all $\theta \in \Theta$*

$$M(\theta) = M(\theta_1) \Rightarrow \theta = \theta_1. \tag{2.37}$$

Essentially the definition means that every dynamic model in the model set is associated with exactly 1 point in the parameter space.

Another necessary condition for consistency is that the data contains a sufficient amount of information. The following definition is common in literature.

Definition 2.9 (Informativity (Ljung, 1999)).
Define the data vector $z(t) = \begin{bmatrix} y^T(t) & u^T(t) \end{bmatrix}^T$ and predictor filter $W(q,\theta) = \begin{bmatrix} H(q,\theta)^{-1} & H(q,\theta)^{-1}G(q,\theta) \end{bmatrix}$ as shorthand notation. The data z(t) is called informative with respect to a model set $\mathcal{M}$ if for any two models $W(q,\theta_1), W(q,\theta_2)$, $\theta_1, \theta_2 \in \Theta$,

$$\mathbb{E}\Big|\Big(W(q,\theta_1) - W(q,\theta_2)\Big)z(t)\Big|^2 = 0 \tag{2.38}$$

implies $W(e^{j\omega},\theta_1) = W(e^{j\omega},\theta_2)$ for almost all ω.

When data is informative for a model set then the predictor filters can be recovered from data. Since the predictor filters are related to G and H in a one-to-one way, the dynamics of G and H can be recovered from data when data is informative. In combination with identifiability this implies that a unique estimate is obtained, i.e. the criterion (2.27) is minimized by a single $\theta \in \Theta$. A sufficient condition for informativity with respect to all model sets is formulated as follows.

> **Theorem 2.10** (Theorem 8.1 in (Ljung, 1999)). *A data set is informative if the spectrum matrix corresponding to $z(t)$ is strictly positive definite for almost all frequencies.*

The theorem shows a condition for informativity of data for any linear time-invariant model. For particular models sets strictly positive definite spectrum may not be necessary, but further analysis on this topic is outside the scope of this thesis.

Now that informativity and identifiability have been defined we can formulate the conditions for consistency. First the conditions for open-loop systems are specified.

> **Proposition 2.11.** *Let data $y(t)$ and $u(t)$ be generated by* (2.23) *in an open-loop situation, i.e. u and v are uncorrelated processes. The estimate $\hat{\theta}_N$ obtained by* (2.27) *with the prediction error described in* (2.33) *is consistent under the following conditions:*
>
> 1. *The model of G is flexible enough, i.e. $\exists\theta_0 \in \Theta$ such that $G(q, \theta_0) = G^0(q)$;*
> 2. *The noise model $H(q, \theta)$ and $G(q, \theta)$ are independently parameterized, i.e. they do not share parameters;*
> 3. *The model set $\mathcal{M}$ is identifiable;*
> 4. *The process u is sufficiently informative.*

When a plant operates in closed-loop, then there is correlation between $u(t)$ and $v(t)$ and a different set of conditions is required for consistency.

> **Proposition 2.12.** *Let data $y(t)$ and $u(t)$ be generated by* (2.23) *in an closed-loop situation, i.e. there is a feedback from y to u. The estimate $\hat{\theta}_N$ obtained by* (2.27) *with the prediction error described in* (2.33) *is consistent under the following conditions:*
>
> 1. *The system is in the model set, i.e. $\exists\theta_0 \in \Theta$ such that both $G(q, \theta_0) = G^0(q)$ and $H(q, \theta_0) = H^0(q)$;*
> 2. *The model set $\mathcal{M}$ is identifiable;*
> 3. *The process $z(t) = \begin{bmatrix} y^T(t) & u^T(t) \end{bmatrix}^T$ is sufficiently informative;*
> 4. *There is at least 1 sample delay in the loop of both the data-generating process and for all models in the model set.*

The three core concepts that are required for consistency are informativity, identifiability, and the data generating system should be in the model set.

2.2.4 Variance

One of the objectives of this thesis is to obtain estimates of minimum variance. The concept variance will be defined in this section. Since the noise $v(t)$ is a random process, the estimate $\hat{\theta}_N$ is a random variable. Estimate $\hat{\theta}_N$ has certain amount of uncertainty, and the uncertainty may be described by the variance. The covariance matrix of a variable x is defined as

$$Cov(x) = \mathbb{E}\left(x - \mathbb{E}(x)\right)\left(x - \mathbb{E}(x)\right)^T. \tag{2.39}$$

It can be shown (Ljung, 1999) that under mild conditions a consistent estimate $\hat{\theta}_N$ is normally distributed with $N \to \infty$, i.e.

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \in As\mathcal{N}(0, P_\theta), \tag{2.40}$$

implying that a consistent estimate $\hat{\theta}_N$ is described by a normal distribution centered at the true parameter with covariance matrix P_θ. Covariance matrix P_θ describes how 'fast' convergence to the true parameter happens, meaning that with small P_θ fewer data points are needed for an accurate estimate compared to high P_θ.

2.3 The MISO direct method

2.3.1 Introduction

In order to identify the network (2.4) some extensions have to be made compared to identification of the open-loop system (2.23). These changes relate to the three core concepts: the data set is larger, the model set is now a network model set, and the criterion has to be adapted. Moreover in the identification we have to account for feedback through the network dynamics, and for the possibility that noise enters at more locations than at one output. Here the direct method for dynamic network identification from (Van den Hof et al., 2013) will be presented in order to show the changes in the identification framework. Since this estimation method is a direct method, it can be related to Maximum Likelihood estimates that minimize the variance, which is a relevant property for answering the research question.

The data set contains all node measurements and all external variables, usually in the form $w(t) \in \mathbb{R}^L$, $r(t) \in \mathbb{R}^K$. This direct method for dynamic network identification is an extension of the direct method for closed-loop identification (Ljung, 1999) that estimates all modules $G_{jl}, l \in \mathcal{N}_j$ in a multi-input-single-output closed-loop setting. If the method is applied to each node j, then all modules in the network can be estimated under certain conditions.

For this direct method it is assumed that data is generated by a data generating system $\mathcal{S}$, that is denoted as the model

$$M^0 = (G^0, R^0, H^0, \Lambda^0). \tag{2.41}$$

The set of candidate models is the network model set as defined in the previous section with some additional restrictions, namely

- References enter the network directly at a single node, without encountering dynamics, i.e. matrix R^0 contains exactly one 1 in every column and is known. This restriction is encoded in the model set, every $M \in \mathcal{M}$ has the same R^0. This also implies that $u_j(t)$ in (2.1) is known.
- Process noise $v_j(t)$ is uncorrelated to other noises in the network, and can be modeled as $v_j(t) = H^0_{jj}(q)e_j(t)$. This restriction is encoded in the model set, every $M \in \mathcal{M}$ has diagonal H and full rank and diagonal Λ.
- Modules G_{jk}, $k \in \mathcal{N}_j$ can have a direct term, but w_j is not part of any algebraic loop.

2.3.2 Definition of the method

The criterion used in prediction error methods is based on the concept of prediction, which essentially is that a good model of node $w_j(t)$ should be able to predict $w_j(t)$ on the basis of the past. Therefore the method makes use of the definition of a one-step-ahead predictor[3] (Van den Hof et al., 2013)

$$\breve{w}_j(t|t-1) := \mathbb{E}\left(w_j(t) \mid w(t-1)^-, w_k(t) \text{ if } G^\infty_{jk} \neq 0, u_j(t)\right), \tag{2.42}$$

where $w(t-1)^-$ is shorthand notation for $w(t-1), w(t-2), \dots, w(t-\infty)$. Essentially the predictor is the part of $w_j(t)$ that can be recovered from past values, and the part that can not be predicted is the innovation $e_j(t)$, i.e.

$$w_j(t) = \breve{w}_j(t|t-1) + e_j(t). \tag{2.43}$$

The innovation $e_j(t)$ is uncorrelated to the predictor $\breve{w}_j(t|t-1)$, otherwise there would be information in the innovation that can be used to predict $w_j(t)$. The predictor can be written as an expression based on the network model that generated the data

$$\breve{w}_j(t|t-1) = w_j(t) - (H^0_{jj})^{-1}(q)\left(w_j(t) - u_j(t) - \sum_{k\in\mathcal{N}_j} G^0_{jk}(q)w_k(t)\right). \tag{2.44}$$

Now the model set can be used to assign parameters to the predictor

$$\breve{w}_j(t|t-1;\theta) = w_j(t) - H^{-1}_{jj}(q,\theta)\left(w_j(t) - u_j(t) - \sum_{k\in\mathcal{N}_j} G_{jk}(q,\theta)w_k(t)\right) \tag{2.45}$$

such that each model is associated with a predictor. The objective then is to find a parameter θ which results in a model that makes good predictions. To this end the difference between measured and predicted w_j is defined as the prediction error

$$\varepsilon(t,\theta) := w_j(t) - \breve{w}_j(t|t-1;\theta). \tag{2.46}$$

[3] The symbol $\breve{w}_j$ is different than the conventionally used predictor symbol $\hat{w}_j$. The reason for this notation will become clear in Chapter 5.

The innovation can not be predicted, and so for the true system the prediction error is equal to the innovation. Now the criterion is supposed to select a model which has a small prediction error. To this end the model is selected that minimizes the power of the prediction error, i.e.

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \theta). \tag{2.47}$$

It can be shown that under some conditions (2.47) leads to consistent estimates.

> **Proposition 2.13** (Proposition 2 in (Van den Hof et al., 2013)). *Estimate* (2.47) *is consistent under the following conditions:*
>
> 1. *Noise v_j is uncorrelated to all reference signals.*
> 2. *Noise v_j is uncorrelated to all other noises that have a path to w_j.*
> 3. *Every loop through w_j has a delay, i.e. there are no algebraic loops through w_j.*
> 4. *The spectral density of $\begin{bmatrix} w_j & w_{n_1} & \dots & w_{n_n} \end{bmatrix}^T$, $n_* \in \mathcal{N}_j$, denoted as $\Phi_{j\mathcal{N}_j}(\omega)$ is positive definite for $\omega \in [-\pi, \pi]$.*
> 5. *The system is in the model set, i.e. there exists a $\theta^0 \in \Theta$ such that $G_{jk}(q, \theta^0) = G^0_{jk}(q)$ for all $k \in \mathcal{N}_j$, and $H_{jj}(q, \theta^0) = H^0_{jj}(q)$.*

This proposition shows that consistent estimates can be obtained from data using the criterion and predictor model as defined above under some conditions. The key conditions described in the proposition are the conditions on correlatedness of the noise, conditions on how feedthrough is modeled, conditions on information content in the data, and conditions on the flexibility of the parameterized model. In the dynamic network model definition we allow for many more networks than the networks that satisfy the conditions of the above proposition. Relaxing conditions under which consistent estimates can be obtained is the objective of the thesis. In particular the following will be relaxed:

- Noise v_j may be correlated with other noises v_i;
- The noise process v may have a rank-reduced spectrum;
- Algebraic loops may be present in the network;
- External variables may now affect the network through a dynamic module;
- A node may have multiple external variables as in-neighbor.

2.3.3 Local identification

For the identification of a single module G_{ji} we may use the MISO direct method with w_j as output and all its in-neighbors as inputs. The identification is then an identification of all modules $G_{jk}, k \in \mathcal{N}_j$ on the basis of input nodes $w_k, k \in \mathcal{N}_j$ and output node w_j. However in (Dankers et al., 2016) it is shown that it is possible to obtain consistent estimates of module G_{ji} when using other nodes than the in-neighbors of w_j as inputs. In that situation the question which nodes to use for the identification of module G_{ji} is apparent. This question is closely related to which nodes must be measured. A further introduction to this topic is provided in Chapter 6.

2.4 Conclusion

In this thesis we are looking for conditions under which efficient estimates of a network may be obtained from data. A number of relaxations in the experimental setup have been made and incoporated into the dynamic network model definition. The defined network model is more flexible than the network models available in literature, and with we may look for the less restrictive conditions for consistency.

In order to have consistent estimates it is necessary that unique models can be obtained. One effect of the flexible network model is that we have to evaluate whether unique network models can be obtained from data. The uniqueness question relates to the identifiability and informativity properties that have been specified for the prediction error method. Conditions under which network models are identifiable are to be investigated.

Identification using the direct method presented in Section 2.3 results in consistent estimates only under the presented restrictions on the experimental setup. When the experimental setup is generalized, for example by allowing correlated noises or algebraic loops, then the direct identification method may no longer result in consistent estimates and must be generalized. Prediction error methods are based on the definition of a predictor, and this predictor needs to be examined and defined for network identification. It is desirable that this new method has favorable variance properties, preferably with the variance at the Cramer-Rao lower bound. A Maximum Likelihood for identification of a dynamic network would lead to such properties. This leads to questions on how to obtain consistent estimates of a dynamic network for such flexible models, and questions regarding the variance of those estimates.

3 Network identifiability

3.1 Introduction

In order to consistently identify modules in a dynamic network we need these modules to be unique minima of the identification criterion. In the defined network model set there is great flexibility in modeling the modules, noise, and external excitations, which may lead to non-uniqueness in the network representation. A-priori knowledge of the experimental setup may be used to restrict the set of models. Then it is to be determined how much the topology, noise, and external excitations need to be restricted such that unique models can be obtained. Based on the papers (Weerts et al., 2018b,c), we are addressing the following question in this chapter.

Under which conditions on the network model set can different network models, or modules, be distinguished from each other on the basis of measured data?

With conditions on the network model set we mean the user choice that is

- the *modeled* presence and location of external excitations,
- the *modeled* correlations between disturbance signals, and
- the *modeled* network topology and module dynamics.

There is an emphasis on the word ”modeled” since the question whether we can distinguish between two models does not depend on any data generating system. The choice for a particular network model set can be based on knowledge of the experimental setup, but this is not strictly necessary. In particular under study are situations that are not commonly addressed in literature, but that are relevant for practical application:

- Disturbance terms v_i are allowed to be correlated over time, and also over node signals, i.e. v_i and v_j, $i \neq j$, can be correlated;
- The vector disturbance process v can have a rank-reduced spectrum, which includes the situation that some individual disturbance terms can be 0 for all time;
- Direct feedthrough terms are allowed in the network modules.

Due to the inclusion of these additional aspects in the dynamic network model there may be non-uniqueness in the network representation, and this will be addressed by an identifiability analysis. Determining restrictions under which network models can be distinguished is the topic of this chapter.

To address the main question the concept of *network identifiability* will be introduced as a property of a parametrized set of network models. Network identifiability will not operate on parameter level, unlike the classical notions introduced in Section 2.2.3. Network identifiability operates on the level of the dynamics, the topology of the network, and on the presence and location of excitations. The first step in this chapter is to start from the definition of a data set, and then determine which information can possibly be recovered from this data set. Identifiability will then be evaluated on the basis of information that can be recovered from data.

When network identifiability is defined, conditions must be obtained under which a model set is network identifiable. In (Gonçalves and Warnick, 2008) conditions for identifiability have been derived for a model named Dynamic Structure Function (DSF). These conditions are to check identifiability on the basis of a nullspace of an identified transfer function matrix. In order to test for identifiability with this method, an identification has to be performed before identifiability can be tested, which is not a practical test. Since the data-generating network is unknown (it has to be identified) we aim to find conditions on the network model set that guarantee identifiability that do not require knowledge of the data-generating network. We want to have conditions that can be verified without the need to know the data generating network or the need to first perform an identification. These conditions can be shown to depend on the modeled topology and experimental setup. When considering network identifiability in a generic sense as is done in (Bazanella et al., 2017), then the conditions may be verified on the basis of available paths and excitations in the network model.

The identifiability analysis relates to certain restrictions on the model set. In (Gonçalves and Warnick, 2008) the investigation is focused on the important aspects

- where external variables enter the network, and
- the modeled topology of the network.

In this chapter some additional phenomena are modeled since we model noise, and we are allowing the presence of modules that are not strictly proper. This leads to some additional aspects of network identifiability to be investigated, namely

- the presence of noise, including possible noise correlations and rank-reduced noise processes, and

- whether it is necessary to have strictly proper modules, or whether feedthrough terms and algebraic loops can be distinguished.

When the objective is to identify a single module, then we no longer need to impose these strong restrictions on the whole network model. Different network models may minimize the cost function, as long as the module of interest is the same in every one of those different network models, such that there is a unique estimate of the module. Network identifiability is then a conservative requirement for the network model, and instead we introduce the concept identifiability of a single module. Restrictions that need to be imposed on the model set can be relaxed compared to network identifiability of the full network model.

The chapter will continue as follows. In Section 3.2 it is shown which information can be obtained from measured data, and the concept network identifiability is defined and discussed. Then in Section 3.3 identifiability issues related to the spectral factorization of noise and direct feedthrough of models are discussed. Conditions under which network models can be distinguished are formulated in Section 3.4 for the situation that every node is excited. For situations where the topology of the network is incorporated in the model set we formulate necessary and sufficient conditions under which network models can be distinguished in Section 3.5. Non-conservative conditions for identifiability of a single module are investigated in Section 3.6. In Section 3.7 identifiability conditions are described in terms of paths in the network. Finally in Section 3.8 the conditions under which the ordering of signals can be determined are shown.

3.2 Definitions

This section will set the basic definitions needed to perform the identifiability analysis. First it must be determined which models to distinguish. Usually in an identification setting a criterion selects a model from the model set, so the models in the model set must be different in some ways such that the criterion can distinguish between the different models.

The question whether in a chosen model set, the models can be distinguished from each other on the basis of measured data, has two important aspects:

- a structural —or identifiability— aspect: is it possible at all to distinguish between models, given the restrictions on the model set, e.g. presence and location of external excitation signals and noise disturbances, and

- a data informativity aspect: are the measured signals informative enough to distinguish between different network models, i.e. is the information present in the data rich enough.

The first (structural) aspect will be referred to as the notion of network identifiability. For consistency of model estimates in an actual identification experiment, it is required that the model set is network identifiable and that the external excitation signals are sufficiently informative. This separation of concepts allows for the study of structural aspects of networks, separate from the particular choice of test signals in identification.

When data is the starting point, we have to determine how the data can be represented such that the identifiability and informativity aspects can clearly be separated. Based on the network equations (2.4), (2.14)-(2.16) the system can be written as

$$\begin{aligned} w &= T_{wr}(q)r(t) + \bar{v}(t), && (3.1)\\ \text{where} \quad \bar{v}(t) &:= T_{we}(q)e(t). && (3.2) \end{aligned}$$

Many identification methods, among which prediction error and subspace identification methods, base their model estimates on second order statistical properties of the measured data. These properties are represented by auto-/cross-correlation functions or spectral densities of the signals w and r. Usually the spectra Φ_w, Φ_{wr} and Φ_r are available as data. Then utilizing the relations of the spectra in (2.19) and (2.20), it is obtained that

$$T_{wr}(e^{i\omega}) = \Phi_{wr}(\omega)\Phi_r^{-1}(\omega), \quad \text{and} \tag{3.3}$$

$$\Phi_{\bar{v}}(\omega) = \Phi_w(\omega) - T_{wr}(e^{i\omega})\Phi_r(\omega)T_{wr}^T(e^{-i\omega}), \tag{3.4}$$

under suitable conditions on the inverse $\Phi_r^{-1}(\omega)$. The transfer function matrix T_{wr} and spectrum $\Phi_{\bar{v}}$ are to be modeled with a parameterized network model. Since w, r are measured, these spectra are written as

$$T_{wr}(e^{i\omega}, \theta) = \Phi_{wr}(\omega)\Phi_r^{-1}(\omega), \quad \text{and} \tag{3.5}$$

$$\Phi_{\bar{v}}(\omega, \theta) = \Phi_w(\omega) - T_{wr}(e^{i\omega}, \theta)\Phi_r(\omega)T_{wr}^T(e^{-i\omega}, \theta). \tag{3.6}$$

Now the objects that can be obtained from the second order properties of the data have been related to the network model, and the distinction between identifiability and informativity aspects can be made. Recovering T_{wr} and $\Phi_{\bar{v}}$ from the available spectra Φ_w, Φ_{wr} and Φ_r will be referred to as the data informativity aspect. Ensuring that models can be distinguished on the basis of T_{wr} and $\Phi_{\bar{v}}$ will be referred to as the identifiability aspect, which is the topic of the remainder of this chapter.

By utilizing (2.15)-(2.16), the parametrized model $M(\theta)$ is related to the objects that can be obtained from data:

$$\begin{aligned} T_{wr}(q, \theta) &:= [I - G(q, \theta)]^{-1}R(q, \theta), \\ \Phi_{\bar{v}}(\omega, \theta) &= [I - G(e^{i\omega}, \theta)]^{-1}H(e^{i\omega}, \theta)\Lambda(\theta) \cdot \\ &\qquad \cdot H(e^{i\omega}, \theta)^*[I - G(e^{i\omega}, \theta)]^{-*}, \end{aligned}$$

where $(\cdot)^*$ denotes complex conjugate transpose. We formalize the reasoning above in the definition of network identifiability that addresses the property that network models are uniquely determined from T_{wr} and $\Phi_{\bar{v}}$.

Definition 3.1 (Network identifiability). *The network model set $\mathcal{M}$ is globally network identifiable at $M_0 := M(\theta_0)$ if for all models $M(\theta_1) \in \mathcal{M}$,*

$$\left.\begin{array}{r} T_{wr}(q,\theta_1) = T_{wr}(q,\theta_0) \\ \Phi_{\bar{v}}(\omega,\theta_1) = \Phi_{\bar{v}}(\omega,\theta_0) \end{array}\right\} \Rightarrow M(\theta_1) = M(\theta_0). \tag{3.7}$$

$\mathcal{M}$ is globally network identifiable if (3.7) holds for all $M_0 \in \mathcal{M}$.

Essentially the definition states that a particular model M_0 is identifiable when it can be distinguished from all other models in the model set $\mathcal{M}$ on the basis of objects that can be determined from 2nd order properties of the data. In an identification setting, the M_0 is unknown and the implication in (3.7) can not be checked directly. Instead of identifiability of a specific M_0, it is sufficient to have identifiability of all models in $\mathcal{M}$, which does not depend on knowledge of M_0. Therefore global network identifiability of a model set $\mathcal{M}$ is a stronger concept than global network identifiability at a particular model. The latter property is considered in (Gonçalves and Warnick, 2008) and (Gevers et al., 2017). In this thesis both properties will be addressed.

We have chosen to use the spectral density $\Phi_{\bar{v}}$ in the definition, rather than its spectral factor as e.g. originally done in Weerts et al. (2015). This is motivated by the objective to include the situation of rank-reduced noise, where $\Phi_{\bar{v}}(\omega,\theta)$ will be singular, and the handling of possible direct feedthrough terms and algebraic loops in the network. This will be further addressed and clarified in Section 3.3.

Remark 3.2. *In Definition 3.1 we consider identifiability of the network dynamics $M = (G, R, H, \Lambda)$. This can simply be generalized to consider the identifiability of a particular network property $f(M)$, by replacing the right hand side of the implication (3.7) by $f(M_1) = f(M_0)$, while f can refer to network properties as e.g. the Boolean topology of the network, the network dynamics in G, a single module G_{ji}, etc.*

Remark 3.3. *The network identifiability definition is formulated for the combined signals w and r. However it is straightforward to formulate the definition for several other situations, e.g. using only signals w as data, or possibly a subset of node signals and a subset of external signals. When only a subset of w and r is considered then the implication* (3.7) *uses the appropriate rows and columns of T_{wr} and $\Phi_{\bar{v}}$ on the left hand side. Note that e.g. in the direct method and joint-io method for closed-loop identification (Ljung, 1999), only the measured signals in w are used as a basis for identifiability studies. In these approaches, excitation signals r may be present, but are not taken into account. This situation can be handled by removing matrix R from the model set.*

Before moving to the formulation of verifiable conditions for network identifiability, an example is presented of a disturbance free network. This is done to illustrate that a model set can be globally identifiable at one model, while not identifiable at another model.

Example 3.4. *Consider the disturbance-free systems $\mathcal{S}_1$, $\mathcal{S}_2$ in Figure 3.1 with $A(q) \neq 0, -1$, and $B(q) \neq 0$, both rational transfer functions. These two networks are described*

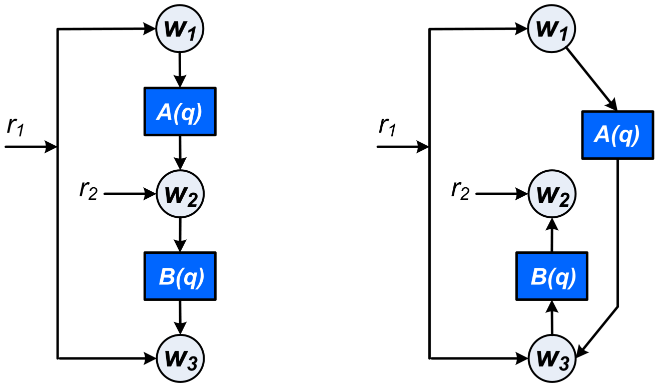


Figure 3.1: Systems $\mathcal{S}_1$ (left) and $\mathcal{S}_2$ (right).

by the transfer functions

$$G_1^0 = \begin{bmatrix} 0 & 0 & 0 \\ A & 0 & 0 \\ 0 & B & 0 \end{bmatrix}, \; G_2^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ A & 0 & 0 \end{bmatrix}, \; R_1^0 = R_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The transfer function matrices $T_{wr}(q)$ related to the networks $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively, are given by:

$$T_1^0(q) = \begin{bmatrix} 1 & 0 \\ A & 1 \\ AB+1 & B \end{bmatrix}, \; T_2^0(q) = \begin{bmatrix} 1 & 0 \\ (A+1)B & 1 \\ A+1 & 0 \end{bmatrix}. \tag{3.8}$$

These transfer functions map the external signals r to the node signals w. We consider the model set $\mathcal{M}(\theta)$ with (omitting arguments q)

$$G(\theta) = \begin{bmatrix} 0 & G_{12}(\theta) & G_{13}(\theta) \\ G_{21}(\theta) & 0 & G_{23}(\theta) \\ G_{31}(\theta) & G_{32}(\theta) & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{3.9}$$

and so $G(\theta)$ is parametrized and R is known and fixed. Since we have a disturbance free system we discard a noise model here, without loss of generality.
In order to investigate whether each of the two systems can be represented uniquely within the model set, we refer to (2.15), and analyze whether the equation

$$T_i^0(q) = [I - G(q,\theta)]^{-1} R(q) \tag{3.10}$$

for $i = 1, 2$ has a unique solution for $G(q,\theta)$. To this end we premultiply (3.10) with $[I - G(q,\theta)]$.
For network $\mathcal{S}_1$ we then obtain the relation (omitting argument q)

$$\begin{bmatrix} 1 & \text{-}G_{12}(\theta) & \text{-}G_{13}(\theta) \\ \text{-}G_{21}(\theta) & 1 & \text{-}G_{23}(\theta) \\ \text{-}G_{31}(\theta) & \text{-}G_{32}(\theta) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A & 1 \\ AB+1 & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.11}$$

Solving the corresponding six equations for the parametrized transfer functions $G_{ij}(\theta)$ shows the following. When combining the two equations related to the first row in the right hand side matrix of (3.11) it follows that $G_{13}(\theta) = G_{12}(\theta) = 0$. Solving the second row leads to $G_{23}(\theta) = 0$ and $G_{21}(\theta) = A$, while solving the third row delivers $G_{31}(\theta) = 0$ and $G_{32}(\theta) = B$. As a result the original system $\mathcal{S}_1$ is uniquely recovered, and so $\mathcal{M}$ is globally network identifiable at $\mathcal{S}_1$.
When applying the same reasoning to network $\mathcal{S}_2$ we obtain

$$\begin{bmatrix} 1 & \text{-}G_{12}(\theta) & \text{-}G_{13}(\theta) \\ \text{-}G_{21}(\theta) & 1 & \text{-}G_{23}(\theta) \\ \text{-}G_{31}(\theta) & \text{-}G_{32}(\theta) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (A+1)B & 1 \\ A+1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.12}$$

Solving this system of equations for the second column on the right hand side leads to $G_{12}(\theta) = G_{32}(\theta) = 0$, while the solution for the first column delivers $G_{13}(\theta) = 0$, $G_{31}(\theta) = A$ and

$$-G_{21}(\theta) + (A+1)B - G_{23}(\theta)(A+1) = 0 \tag{3.13}$$

or equivalently $G_{21}(\theta) = (A+1)(B - G_{23}(\theta))$. This shows that not only $G_{21}(\theta) = 0$, $G_{23}(\theta) = B$ is a valid solution, but actually an infinite number of solutions exists. As a result $\mathcal{M}$ is not globally network identifiable at $\mathcal{S}_2$. An interpretation is that in $\mathcal{S}_2$ the contributions from w_1 and w_3 both solely depend on r_1 making them indistinguishable, which is reflected in the modelled transfer function matrix $R(q)$.

3.3 Conditions on feedthrough and noise correlation

3.3.1 Problem description

In this section we show that it is impossible to distinguish between feedthrough in G and instantaneous correlations in process noise described by Λ on the basis of only the spectrum $\Phi_{\bar{v}}$. For this reason we provide three sets of conditions, presented as restrictions on the network model set, under which we are able to uniquely recover the feedthrough and Λ.

In papers on identification in dynamic networks, different assumptions are made on the presence of direct feedthrough terms G^{∞} and noise correlation Λ. For example in (Van den Hof et al., 2013) it is assumed that Λ is diagonal and that there are no algebraic loops in the network, while in (Gonçalves and Warnick, 2008) it is assumed that all modules are strictly proper.

Here we consider a general situation where there can be algebraic loops and there may be correlation between the noises. In this situation the question is whether a distinction can be made between noise correlation and direct feedthrough terms. This is illustrated by an example.

Example 3.5. *In this example there are two networks, shown in Figure 3.2, which have $cov(e) = I$. The spectrum of $w = \bar{v}$ is in both systems exactly the same*

$$\Phi_w = \begin{bmatrix} 1 & a \\ a & 1+a^2 \end{bmatrix}. \tag{3.14}$$

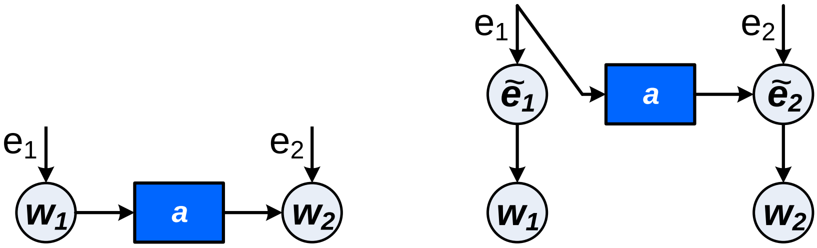


Figure 3.2: Systems $\mathcal{S}_1$ (left) and $\mathcal{S}_2$ (right).

In the left system, the correlation between w_1 and w_2 is explained with a module, while in the right system the correlation is due to correlation of noise process $\tilde{e}_1$ with $\tilde{e}_2$. This leads to two different decompositions of the spectrum

$$\Phi_w = \underbrace{\begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}^{-1}}_{(I-G_1)^{-1}} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\Lambda_1} \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}^{-1} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1}}_{(I-G_2)^{-1}} \underbrace{\begin{bmatrix} 1 & a \\ a & 1+a^2 \end{bmatrix}}_{\Lambda_2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1}. \tag{3.15}$$

When both these models are present in the model set $\mathcal{M}$, then it is impossible to distinguish between the models on the basis of only the spectrum $\Phi_{\bar{v}}$.

Based on the example, the central question becomes which restrictions should be incorporated into a model set such that the feedthrough can be distinguished from the noise correlation. These restrictions may be imposed on either the feedthrough terms of G, or on the noise correlations captured in Λ, such that the following implication is satisfied

$$\Phi_{\bar{v}}(\omega,\theta_1) = \Phi_{\bar{v}}(\omega,\theta_0) \Rightarrow \begin{cases} G^\infty(\theta_1) = G^\infty(\theta_0) \\ \Lambda(\theta_1) = \Lambda(\theta_0) \end{cases}. \tag{3.16}$$

Here we note that the decomposition of the spectrum

$$\Phi_{\bar{v}}(\omega,\theta) = T_{we}(e^{i\omega})\Lambda T_{we}^T(e^{-i\omega}), \tag{3.17}$$

with $T_{we} = (I-G)^{-1}H$ can only be unique when implication (3.16) holds. We have seen in Example 3.5 that implication (3.16) does not hold for all network model sets. The spectrum $\Phi_{\bar{v}}$ is uniquely be decomposed whenever the following implication holds

$$\left.\begin{matrix} T_{wr}(q,\theta_1) = T_{wr}(q,\theta_0) \\ \Phi_{\bar{v}}(\omega,\theta_1) = \Phi_{\bar{v}}(\omega,\theta_0) \end{matrix}\right\} \Rightarrow \begin{cases} T_{wr}(q,\theta_1) = T_{wr}(q,\theta_0) \\ T_{we}(q,\theta_1) = T_{we}(q,\theta_0) \\ \Lambda(\theta_1) = \Lambda(\theta_0) \end{cases}, \tag{3.18}$$

where we include T_{wr} such that we may make explicit use of external variables. Since external variables have a known spectrum they may be used to uniquely determine the feedthrough of G. In the remainder of this section we provide three sets of conditions for which (3.18) is satisfied. These conditions cover three different situations by specifying particular assumptions on the presence/absence of delays in the modules in the networks, by allowing certain correlations in the noise model, and by making explicit use of external variables.

For situations where (3.18) is holds, the implication (3.7) of Definition 3.1 can be reformulated into a condition on the network transfer functions $T(q,\theta)$ and $\Lambda(\theta)$. When the implication (3.18) holds, then network identifiability is shown when implication

$$T(q,\theta_1) = T(q,\theta_0) \Rightarrow (G(\theta_1), R(\theta_1), H(\theta_1)) = (G(\theta_0), R(\theta_0), H(\theta_0)) \tag{3.19}$$

holds. In order to provide conditions for network identifiability, in this section we will provide conditions such that (3.18) holds. Then under those conditions network identifiability can be formulated for the implication (3.19), for which we provide conditions in later sections.

Remark 3.6. *In Gonçalves and Warnick (2008) the transfer function T has been used as a basis for dynamic structure reconstruction. The fact that the network transfer function T is the object that can be uniquely identified from data, has been analyzed in Weerts et al. (2015) for the situation that $p = L$ with diagonal $\Lambda(\theta)$, and no algebraic loops in the networks. This has been the motivation in Weerts et al. (2015) to use the condition (3.19) as a definition of network identifiability. In the extended situation with possible algebraic loops, rank-reduced noise, and without restrictions on noise correlations, we have to formulate conditions such that both* (3.18) *and* (3.19) *are satisfied in order to guarantee network identifiability.*

3.3.2 Decompositions of the noise spectrum

First we consider the situation that all modules in the network are strictly proper, i.e. $\lim_{z\to\infty} G(z) = 0$. When noise is also full rank, then $(I-G)^{-1}H$ is monic, such that it becomes rather straightforward to make a unique spectral decomposition of $\Phi_{\bar{v}}$. The following proposition covers the full rank as well as the rank-reduced situation.

Proposition 3.7. *Consider a network model set $\mathcal{M}$, and define $T(q,\theta)$ being the parameterized version of the network transfer function $T(q)$ (2.14). If*

$$G^{\infty}(\theta) := \lim_{z\to\infty} G(z,\theta) = 0 \textit{ for all } \theta \in \Theta, \tag{3.20}$$

then implication (3.18) *holds, and condition (3.7) in Definition 3.1 of network identifiability is equivalently formulated as* (3.19)*.*

Proof. Provided in Appendix 3.10.1. ■

Note that the above result is valid for both full-rank ($p = L$) and reduced-rank ($p < L$) noise processes. In the rank-reduced case T_{we} and H become non-square of dimension $L \times p$. Additionally there are no restrictions on $\Lambda(\theta)$, e.g. it is not restricted to being diagonal.

The second situation that is addressed is where modules are allowed to have feedthrough terms, such as was done in Weerts et al. (2015). For this situation we have to show that that (3.7) in Definition 3.1 of network identifiability is equivalently formulated as (3.19). Feedthrough terms make $(I-G)^{-1}$ a non-monic transfer matrix, meaning that we must be careful with factorization of the spectrum $\Phi_{\bar{v}}$. It can be shown (see Dankers (2014)) that there are no algebraic loops in a network if and only if there exists a permutation matrix Π, such that $\Pi^T G^{\infty} \Pi$ is upper triangular.

Proposition 3.8. *Consider a network model set $\mathcal{M}$, and define $T(q,\theta)$ being the parameterized version of the network transfer function $T(q)$ (2.14). If*

1. *There exists a permutation matrix Π such that for all $\theta \in \Theta$,*

$$\Pi^T G^\infty(\theta)\Pi \tag{3.21}$$

is upper triangular, and

2. *$\Phi_v^\infty(\theta) := H^\infty(\theta)\Lambda(\theta)H^\infty(\theta)^T$ is diagonal for all $\theta \in \Theta$,*

then implication (3.18) *holds, and condition (3.7) in Definition 3.1 of network identifiability is equivalently formulated as* (3.19).

Proof. Provided in Appendix 3.10.2. ■

The interpretation of condition *1.* seems to be that there is no algebraic loop in the parametrized model set. However condition *1.* is actually stronger than just absence of algebraic loops, it also implies that the location of feedthrough terms is the same in every model that is in the model set. This means that a user has to choose, based on some a-priori knowledge, which modules are allowed to have a direct term, and which have a delay.

When comparing Proposition 3.8 to Proposition 3.7 we make the following observation. The ability to estimate more flexible correlations between the white noise processes ($\Phi_v^\infty(\theta)$ is not constrained in Proposition 3.7, while being diagonal in Proposition 3.8), and this is traded against the ability to handle direct feedthrough terms in the modules (Proposition 3.8). It should be noted that the above results hold true for any selection of excitation signals r that are present.

3.3.3 The situation of algebraic loops

There are some situations where the modeling of algebraic loops can not be avoided. One example is when we are trying to detect the location of feedthrough terms in all of the modules. With such a model the propositions 3.7 and 3.8 are not applicable, and the results need to be extended.

The results of Proposition 3.7 and 3.8 have been derived based on conditions that guarantee that the transfer function T_{we} and correlation matrix Λ are uniquely determined from the noise spectrum $\Phi_{\bar{v}}$. By making use of external variables r, and by incorporating specific conditions on T_{wr}, more generalized situations can be handled, even including the situation of having algebraic loops in the network. We will follow a reasoning where the transfer function T_{wr} will be required to uniquely determine the feedthrough term G^∞, and –as a result— also the noise covariance matrix Λ.

To this end we consider the matrices $T_{wr}^\infty(\theta)$, $R^\infty(\theta)$ and $G^\infty(\theta)$ that contain direct feedthrough terms of $T_{wr}(q,\theta)$, $R(q,\theta)$ and $G(q,\theta)$. Suppose that row j of $(I-G^\infty(\theta))$ has α_j parameterized elements, and row j of $R^\infty(\theta)$ has β_j parameterized elements.

We define the $L \times L$ permutation matrix P_j and the $K \times K$ permutation matrix Q_j such that all parametrized entries in the considered row of $(I - G^{\infty}(\theta))P_j$ are gathered on the left hand side, and all parametrized entries in the considered row of $R^{\infty}(\theta)Q_j$ are gathered on the right hand side, i.e.

$$(I - G^{\infty}(\theta))_{j\star}P_j = \begin{bmatrix}(I - G^{\infty}(\theta))^{(1)}_{j\star} & (I - G^{\infty})^{(2)}_{j\star}\end{bmatrix} \tag{3.22}$$

$$R^{\infty}(\theta)_{i\star}Q_j = \begin{bmatrix}R^{\infty}{}^{(1)}_{j\star} & R^{\infty}{}^{(2)}_{j\star}(\theta)\end{bmatrix} \tag{3.23}$$

with $(\cdot)_{j\star}$ indicating the j-th row of a matrix.

Next we define the matrix $\check{T}^{\infty}_j(\theta)$ of dimension $\alpha_j \times (K - \beta_j)$ as the submatrix of $T^{\infty}_{wr}(\theta)$ that is constructed by taking the row numbers that correspond to the columns of $G^{\infty}(\theta)_{j\star}$ that are parametrized, and by taking the column numbers that correspond to the columns of $R^{\infty}(\theta)$ that are not parametrized. This is formalized by

$$\check{T}^{\infty}_j\left(T_{wr}(\theta)\right) := \begin{bmatrix}I_{\alpha_j} & 0\end{bmatrix} P_j^{-1} T^{\infty}_{wr}(\theta) Q_j \begin{bmatrix}I_{K-\beta_j} \\ 0\end{bmatrix}. \tag{3.24}$$

We can now formulate the following identifiability result for the situation that algebraic loops are allowed in the network.

> **Proposition 3.9.** *Consider a network model set $\mathcal{M}$, and define $T(q,\theta)$ being the parameterized version of the network transfer function $T(q)$ (2.14). If for all $\theta \in \Theta$:*
>
> - *each row of $\begin{bmatrix}G^{\infty}(\theta) & R^{\infty}(\theta)\end{bmatrix}$ has at most K parameterized elements, and*
> - *for each $j = 1, \cdots L$, the matrix $\check{T}^{\infty}_j\left(T_{wr}(\theta)\right)$ has full row rank for all $\theta \in \Theta$,*
>
> *then implication* (3.18) *holds, and condition (3.7) in Definition 3.1 of network identifiability is equivalently formulated as (3.19).*

Proof. Provided in Appendix 3.10.3. ■

In the above proposition, conditions are formulated under which the transfer function T_{wr} will uniquely determine the direct-feedthrough term G^{∞} and —as a result thereof— also the noise covariance matrix Λ.

3.4 Identifiability conditions for full excitation

It has been shown that for three situations the essential condition for global network identifiability can be equivalently formulated in the expression (3.19) on the basis of T. In order to show conditions for network identifiability we have to provide conditions such that implication (3.19) holds. Here we investigate conditions under which the implication (3.19) is satisfied for the situation that $K + p \geq L$, i.e. where we have at least as many external signals as nodes available. This leads to sufficient conditions for global network identifiability on the basis of where external excitation and noise enter the network, but that are not dependent on the particular structure of the network as present in G. When conditions are not dependent on the particular structure of the modeled G, then these conditions can be satisfied even when no a-priori knowledge of the structure of G is included in the model set.

Theorem 3.10. *Let $\mathcal{M}$ be a network model set for which the conditions of one of the Propositions 3.7-3.9 are satisfied, and where all modules in G may be parameterized. Let $U = [H(q) \;\; R(q)]$ as defined in* (2.17). *Then*

(a) $\mathcal{M}$ is globally network identifiable at $M(\theta_0)$ if there exists a nonsingular and parameter-independent transfer function matrix $Q \in \mathbb{R}^{(K+p)\times(K+p)}(z)$ such that

$$U(q,\theta)Q(q) = \begin{bmatrix} D(q,\theta) & F(q,\theta) \end{bmatrix} \tag{3.25}$$

with $D(\theta) \in \mathbb{R}^{L\times L}(z)$, $F(\theta) \in \mathbb{R}^{L\times(p+K-L)}(z)$, and D diagonal and full rank for all $\theta \in \Theta_0$ with

$$\Theta_0 := \{\theta \in \Theta \mid T(q,\theta) = T(q,\theta_0)\}.$$

(b) If in part (a) the diagonal and full rank property of $D(q,\theta)$ is extended to all $\theta \in \Theta$, then $\mathcal{M}$ is globally network identifiable.

Proof. Provided in Appendix 3.10.4. ■

Expression (3.25) is basically equivalent to a related result in (Gonçalves and Warnick, 2008), where a deterministic reconstruction problem is considered on the basis of a network transfer function, however without considering (non-measured) stochastic disturbance signals. Note that the condition can be interpreted as the possibility to give $U(q,\theta)$ a leading diagonal matrix by column operations. There is an implicit requirement in the theorem that U has full row rank, and therefore it does not apply to the case of Example 3.4. The situation of a rank-reduced matrix U will be considered in Section 3.5.

Example 3.11. *Suppose we model correlated noise by having off-diagonal terms in H, in the model set $\mathcal{M}(\theta)$ with*

$$G = \begin{bmatrix} 0 & G_{12}(\theta) & G_{13}(\theta) \\ G_{21}(\theta) & 0 & G_{23}(\theta) \\ G_{31}(\theta) & G_{32}(\theta) & 0 \end{bmatrix},$$

$$H = \begin{bmatrix} H_{11}(\theta) & H_{12}(\theta) & 0 \\ H_{21}(\theta) & H_{22}(\theta) & 0 \\ 0 & 0 & H_{33}(\theta) \end{bmatrix}, \; R = \begin{bmatrix} R_{11}(\theta) & 0 \\ 0 & R_{22}(\theta) \\ 0 & 0 \end{bmatrix},$$

where $R_{11}(\theta), R_{22}(\theta) \not\equiv 0$, and $H(\theta)$ monic. Then a simple permutation matrix Q can be found to create $U(q,\theta)Q = \begin{bmatrix} D(q,\theta) & F(q,\theta) \end{bmatrix}$ with

$$D(q,\theta) = \operatorname{diag}(\begin{bmatrix} R_{11}(\theta) & R_{22}(\theta) & H_{33}(\theta) \end{bmatrix})$$

and by Theorem 3.10 the model set is globally network identifiable. If external excitation signals r would have been absent, identifiability can not be guaranteed according to Theorem 3.10 because the off-diagonal terms in the noise model would prevent the existence of a permutation matrix Q that can turn the noise model into a diagonal form. If the process noises at the first two nodes are uncorrelated, they can be modelled with $H_{21}(\theta) \equiv H_{12}(\theta) \equiv 0$, and the diagonal H directly implies global network identifiability, irrespective of the presence of external excitation signals.

One of the important consequences of Theorem 3.10 is formulated in the next corollary.

> **Corollary 3.12.** *Subject to the conditions in Theorem 3.10, a network model set $\mathcal{M}$ is globally network identifiable if every node signal in the network is excited by either an external excitation signal r or a noise signal v, that is uncorrelated with the excitaton/noise signals on the other nodes.*

The situation described in the Corollary corresponds to $U(q,\theta)$ having a single parameterized entry in every row and every column, and thus implies that $U(q,\theta)$ can be permuted to a diagonal matrix. Uncorrelated excitation can come from noise or external variables. Note that the result of Theorem 3.10 can be rather conservative, as it does not take account of any possible structural conditions in the matrix $G(q,\theta)$. Additionally the result does not apply to the situation where $U(q,\theta)$ is not full row rank, i.e. when the number of external variables plus the rank of the process noise is smaller than the number of nodes, as in that case U can never be transformed to having a leading diagonal by column operations. This is e.g. the case in Example 3.4. Both structural constraints and possible reduced row rank of $U(q,\theta)$ will be further considered in Section 3.5. Now some illustrative examples are presented that originate from Weerts et al. (2016a).

Example 3.13 (Closed-loop system)**.** *One of the very simple examples to which the results above apply is the situation of a single-loop feedback system, with a disturbance signal on the process output, and a reference input at the process input (controller output), see Figure 3.3. The process output y will take the role of node variable w_a, while the process input u will be represented by the (noise-free) w_b. When parametrizing process $G(q,\theta)$ and controller $C(q,\theta)$, as well as noise model $v(t) = H_a(q,\theta)e(t)$ and the fixed reference filters $R_a(q) = 0$, $R_b(q) = 1$, it appears that the essential identifiability result of Theorem 3.10 is reflected by the matrix*

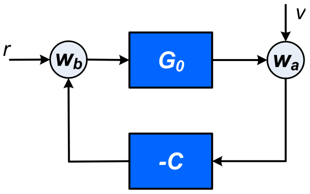


Figure 3.3: Classical closed-loop configuration.

$$U(q,\theta) = \begin{bmatrix} H_a(q,\theta) & 0 \\ 0 & 1 \end{bmatrix}.$$

This matrix is square and equal to the diagonal matrix D in the theorem. Since it is square we have that matrix F will have dimension 2×0. The conditions of Theorem 3.10 are satisfied with $Q = I$, and therefore the closed-loop system is globally network identifiable. This implies that consistent estimates of G_0 and C can be obtained, when identified simultaneously.

In our current setting we consider the simultaneous identification of all modules in the network. In the classical direct method of closed-loop identification, one typically parametrizes the plant model G, but not the controller C, implying that only part of the network is identified. This can lead to questions of identifiability of part of a network (rather than of the full network). The analysis of such a question can fit into the general setting of Definition 3.1 by considering the network property $f(M) = G$, as meant in Remark 3.2, and this will be addressed in a later section.

Example 3.14 (Network example)**.** *In this example we analyze the 5 node network of Figure 3.4 where the noises on nodes 1 and 2 are correlated. The nodes are labeled such that the last two are noise-free. In this example no knowledge on the structure of G is included in the network model set. Process noise will be modeled according to*

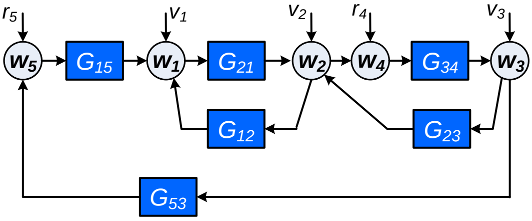


Figure 3.4: 5 node network.

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \underbrace{\begin{bmatrix} H_{11}(q,\theta) & H_{12}(q,\theta) & 0 \\ H_{21}(q,\theta) & H_{22}(q,\theta) & 0 \\ 0 & 0 & H_{33}(q,\theta) \end{bmatrix}}_{H_a(q,\theta)} \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix}.$$

The elements H_{21} and H_{12} are present to allow for modeling correlation between the process noises v_1 and v_2, while v_3 is modeled independently from these two signals. As the external excitation signals r_4 and r_5 directly affect the two corresponding node signals, without a dynamic transfer, the corresponding R matrices are not parameterized but fixed to 1. This leads to a matrix $U(q,\theta)$ constructed as

$$U(q,\theta) = \begin{bmatrix} H_{11}(q,\theta) & H_{12}(q,\theta) & 0 & 0 & 0 \\ H_{21}(q,\theta) & H_{22}(q,\theta) & 0 & 0 & 0 \\ 0 & 0 & H_{33}(q,\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The condition of Theorem 3.10 is now checked by attempting to diagonalize the matrix $U(q,\theta)$ by postmultiplication with some filter $Q(q)$ which does not depend on θ. Due to *the correlated noise it is not possible to diagonalize the matrix in this way. Note that by adding external excitations to nodes* 1 *and* 2*, leading to the addition of fixed unit vector columns in* $U(q,\theta)$*, we can make the model set globally network identifiable.*

3.5 Identifiability for known network topology

When the topology of $G(q,\theta)$, or a part of it, is known, or when there are fewer external variables plus noises, as in Example 3.4, the result of Theorem 3.10 is conservative and/or even does not apply. Structure restrictions in $G(q,\theta)$ are typically represented by fixing some modules, possibly to 0, on the basis of assumed prior knowledge. For these cases of structure restrictions, in (Gonçalves and Warnick, 2008) necessary and sufficient conditions have been formulated for satisfying global network identifiability at a particular model M_0. The conditions are formulated in terms of nullspaces that cannot be checked without knowledge of M_0. Since we are most interested in global identifiability of a full model set, rather than in a particular model, we will further elaborate and generalize these conditions and present them in a form where these conditions can be checked without a-priori knowledge of the network dynamics.

First we need to introduce some notation. In line with the reasoning in Section 3.3.3, we suppose that each row j of $G(\theta)$, has α_j parameterized transfer functions, and row j of $U(\theta)$ has β_j parametrized transfer functions, and we define the $L\times L$ permutation matrix P_j, and the $(K+p)\times(K+p)$ permutation matrix Q_j, such that all parametrized entries in the considered row of $(I-G(q,\theta))P_j$ are gathered on the left hand side, and all parametrized entries in the considered row of $U(q,\theta)Q_j$ are gathered on the right hand side, i.e.

$$(I-G(\theta))_{j\star}P_j = \begin{bmatrix}(I-G(\theta))^{(1)}_{j\star} & (I-G)^{(2)}_{j\star}\end{bmatrix} \tag{3.26}$$

$$U(\theta)_{j\star}Q_j = \begin{bmatrix}U^{(1)}_{j\star} & U(\theta)^{(2)}_{j\star}\end{bmatrix} \tag{3.27}$$

Next we define the transfer matrix $\check{T}_j(q,\theta)$ of dimension $\alpha_j\times(K+p-\beta_j)$, as the submatrix of $T(q,\theta)$ that is constructed by taking the row numbers that correspond to the columns of $G(q,\theta)_{j\star}$ that are parametrized, and by taking the column numbers that correspond to the columns of $U(q,\theta)$ that are not parametrized. This is formalized by

$$\check{T}_j\left(T(q,\theta)\right) := \begin{bmatrix}I_{\alpha_j} & 0\end{bmatrix} P_j^{-1}T(q,\theta)Q_j\begin{bmatrix}I_{K+p-\beta_j}\\ 0\end{bmatrix}. \tag{3.28}$$

The following theorem now specifies necessary and sufficient conditions for the central identifiability condition (3.19).

Theorem 3.15. *Let* $\mathcal{M}$ *be a network model set for which the conditions of one of the Propositions 3.7-3.9 are satisfied, and that additionally satisfies the following properties:*

a. *Every parametrized entry in the model* $\{M(z,\theta),\theta\in\Theta\}$ *covers the set of all proper rational transfer functions;*

> *b. All parametrized transfer functions in the model $M(z,\theta)$ are parametrized independently (i.e. there are no common parameters).*
>
> *Let $U = [H(q) \;\; R(q)]$ as defined in (2.17). Then*
>
> *1. $\mathcal{M}$ is globally network identifiable at $M(\theta_0)$ if and only if*
>
> - *each row j of the transfer function matrix $\begin{bmatrix}G(\theta) & U(\theta)\end{bmatrix}$ has at most $K+p$ parameterized entries, and*
> - *for each j, $\check{T}_j\left(T(q,\theta_0)\right)$ defined by (3.28) has full row rank.*
>
> *2. $\mathcal{M}$ is globally network identifiable if and only if*
>
> - *each row j of the transfer function matrix $\begin{bmatrix}G(\theta) & U(\theta)\end{bmatrix}$ has at most $K+p$ parameterized entries, and*
> - *for each j, $\check{T}_j\left(T(q,\theta)\right)$ defined by (3.28) has full row rank for all $\theta \in \Theta$.*

Proof. Provided in Appendix 3.10.5. ■

Condition *a.* states that a parameterized model is not restricted in model order or dynamics, which means the transfer function is modeled in a non-parametric way. The reason for using the non-parametric transfer functions is that network identifiability is defined on the structural level, and not the parameter level where classical identifiability notions are defined. In case a model set with restrictions on the transfer functions such as a limited order or shared parameters, then these restrictions may relax the necessary conditions for network identifiability. Sufficiency of the conditions does however not depend on the parameterization of the transfer functions. This means that the theorem is applicable to parametric models

The condition on the maximum number of parametrized entries in the transfer function matrix, reflects a condition that the number of parametrized transfers that map into a particular node, should not exceed the total number of excitation signals plus white noise signals that drive the network. The check on the row rank of matrices $\check{T}_j$ is an explicit way to check the related nullspace condition in (Gonçalves and Warnick, 2008). The assumption (a.) in the theorem, refers to the situation that we do not restrict the model class to any finite dimensional structure, but that we consider the situation that could be represented by a non-parametric identification of all module elements. Essentially we are separating the network identifiability concept from the parameter identifiability concept that was presented in Section 2.2.3.

Remark 3.16. *The condition on the maximum number of parametrized entries per row in the parametrized matrix seems closely related to a similar condition for structural identifiability of (polynomial) ARMAX systems, as formulated in Theorem 2.7.1 of Hannan and Deistler (1988).*

The results of this Section can be applied to Example 3.4.

Example 3.17 (Example 3.4 continued)**.** *In Example 3.4 a model set has been defined with $U = R$ not full row rank, and hence Theorem 3.10 is not suitable for checking its* *network identifiability. Now with the introduction of necessary and sufficient conditions in Theorem 3.15 we can evaluate the network identifiability property of the model set in Example 3.4 easily. Consider a model set with G and R as defined in (3.9), without noise model (i.e. $p = 0$) such that $U = R$, and satisfying assumptions (a.) and (b.) of Theorem 3.15. Global network identifiability at $\mathcal{S}_1$ and $\mathcal{S}_2$ is evaluated by checking the two conditions of Theorem 3.15. First it is easily verified that $\begin{bmatrix} G(\theta) & U \end{bmatrix}$ has at most $2 = K + p$ parameterized transfer functions on each row. The second condition is checked by evaluating the rank of the appropriate sub-matrices defined in (3.28) on the basis of the T-matrices for $\mathcal{S}_1$ and $\mathcal{S}_2$ given in (3.8).*

For $\mathcal{S}_1$ we need to check the conditions for rows 1-3 accordingly. Then by considering (3.11) we can determine $\check{T}_j\left(T(q,\theta_1)\right)$ as appropriate submatrices of $T(q,\theta_1)$. For all rows j, $Q_j = I$, since U is not parametrized, and so we need to consider all columns of $T(q,\theta_1)$. For $j = 1$, $\check{T}_j\left(T(q,\theta_1)\right)$ is defined by selecting the second and third row of $T(q,\theta_1)$, corresponding with the columns of parametrized elements in $G_{1\star}(q,\theta_1)$, i.e.

$$\check{T}_1\left(T(q,\theta_1)\right) = \begin{bmatrix} A & 1 \\ AB+1 & B \end{bmatrix};$$

while for $j = 2$ we need to select rows one and three, and for $j = 3$ rows one and two, corresponding with the columns of parametrized elements in $G_{2\star}(q,\theta_1)$ and $G_{3\star}(q,\theta_1)$, respectively, leading to

$$\check{T}_2\left(T(q,\theta_1)\right) = \begin{bmatrix} 1 & 0 \\ AB+1 & B \end{bmatrix}, \quad \check{T}_3\left(T(q,\theta_1)\right) = \begin{bmatrix} 1 & 0 \\ A & 1 \end{bmatrix}.$$

Since all three $\check{T}$-matrices are full row-rank, since we assumed that $B \neq 0$, the conditions for global network identifiability at $\mathcal{S}_1$ are satisfied which verifies the conclusion of Example 3.4.

For $\mathcal{S}_2$ a similar check needs to done on the basis of (3.12), leading to

$$\check{T}_2\left(T(q,\theta_2)\right) = \begin{bmatrix} 1 & 0 \\ A+1 & 0 \end{bmatrix}$$

which obviously does not have full rank, confirming that the model set is not globally network identifiable at $\mathcal{S}_2$.

If we would restrict the model set to satisfy $G_{21}(\theta) = 0$, it can simply be verified that the conditions for global network identifiability at $\mathcal{S}_2$ are satisfied, which is confirmed by the analysis in (3.13).

3.6 Identifiability of a single module

The next step is to relax the definition and conditions from the previous sections such that uniqueness of a single module can be guaranteed. To this end we formalize identifiability of particular properties of M as suggested in Remark 3.2. First we define identifiability of a row of M, in order to evaluate identifiabilty around a certain node in a network, after which identifiability of a particular module is treated.

Definition 3.18. *Row j of network model set $\mathcal{M}$ is globally network identifiable at $M_0 := M(\theta_0)$ if for all models $M(\theta_1) \in \mathcal{M}$,*

$$\left.\begin{array}{c} T_{wr}(q,\theta_1) = T_{wr}(q,\theta_0) \\ \Phi_{\bar{v}}(\omega,\theta_1) = \Phi_{\bar{v}}(\omega,\theta_0) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} G_{j\star}(q,\theta_1) = G_{j\star}(q,\theta_0) \\ R_{j\star}(q,\theta_1) = R_{j\star}(q,\theta_0) \\ H_{j\star}(q,\theta_1) = H_{j\star}(q,\theta_0) \end{array}\right. . \tag{3.29}$$

Row j of network model set $\mathcal{M}$ is globally network identifiable if (3.29) holds for all $M_0 \in \mathcal{M}$.

Similar to the reasoning of Section 3.3 we can reason that restrictions on feedthrough terms and noise correlation are needed in order to ensure identifiability.

Corollary 3.19. *Let $\mathcal{M}$ be a network model set for which the conditions of one of the Propositions 3.7-3.9 are satisfied, then implication* (3.18) *holds, and condition (3.29) in Definition 3.18 of network identifiability of row j is equivalently formulated as*

$$T(q,\theta_1) = T(q,\theta_0)\} \Rightarrow \left\{\begin{array}{l} G_{j\star}(q,\theta_1) = G_{j\star}(q,\theta_0) \\ R_{j\star}(q,\theta_1) = R_{j\star}(q,\theta_0) \\ H_{j\star}(q,\theta_1) = H_{j\star}(q,\theta_0) \end{array}\right. . \tag{3.30}$$

The conditions in Theorem 3.15 are formulated independently for each row, so it is straightforward to obtain conditions under which a specific row of $\mathcal{M}$ is identifiable.

Corollary 3.20. *Let $\mathcal{M}$ be a network model set as defined in Theorem 3.15, and let $\check{T}_j\left(T(q,\theta)\right)$ be defined by (3.28), then:*

1. *Row j of network model set $\mathcal{M}$ is globally network identifiable at $M(\theta_0)$ if and only if*

   i) *row j of the transfer function matrix $\begin{bmatrix} G(\theta) & U(\theta) \end{bmatrix}$ has at most $K+p$ parameterized entries, and*

   ii) *$\check{T}_j\left(T(q,\theta_0)\right)$ has full row rank.*

2. *Row j of network model set $\mathcal{M}$ is globally network identifiable if and only if* i) *holds and* ii) *holds for all $\check{T}_j\left(T(q,\theta)\right), \theta \in \Theta$.*

When we are interested in uniquely distinguishing one specific module, then the identifiability conditions of a row of $\mathcal{M}$ are conservative. It is possible that a module is identifiable, even when other modules of that row are not, which is illustrated by the following example.

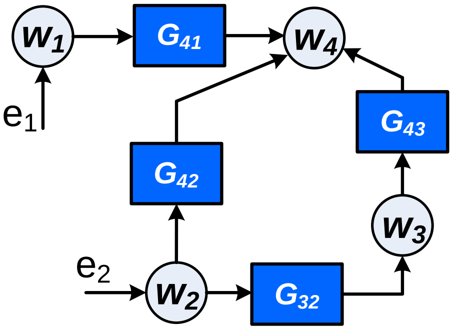


Figure 3.5: Example network model where some modules are identifiable.

Example 3.21. *Consider a set of network models of the topology shown in Figure 3.5, described by*

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & G_{32} & 0 & 0 \\ G_{41} & G_{42} & G_{43} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \tag{3.31}$$

where all modules G_{ji} are parameterized. The response of the node variables is given by $w = Te$ with

$$T = (I - G)^{-1} H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & G_{32} \\ G_{41} & G_{42} + G_{32}G_{43} \end{bmatrix}. \tag{3.32}$$

From T module G_{41} can directly be determined, but the other modules G_{42} and G_{43} on row 4 can not. This is because node 4 has three modules, and there are only two available excitations.

Identifiability of a specific module is defined next.

> **Definition 3.22.** *Module G_{ji} of network model set $\mathcal{M}$ is globally network identifiable at $M_0 := M(\theta_0)$ if for all models $M(\theta_1) \in \mathcal{M}$,*
>
> $$\left. \begin{array}{r} T_{wr}(q, \theta_1) = T_{wr}(q, \theta_0) \\ \Phi_{\bar{v}}(\omega, \theta_1) = \Phi_{\bar{v}}(\omega, \theta_0) \end{array} \right\} \Rightarrow \{G_{ji}(q, \theta_1) = G_{ji}(q, \theta_0)\}. \tag{3.33}$$
>
> *Module G_{ji} of network model set $\mathcal{M}$ is globally network identifiable if (3.33) holds for all $M_0 \in \mathcal{M}$.*

Also for this definition we can directly formulate conditions on feedthrough terms and instantaneous noise correlations.

Corollary 3.23. *Let $\mathcal{M}$ be a network model set for which the conditions of one of the Propositions 3.7-3.9 are satisfied, then implication* (3.18) *holds, and condition (3.33) in Definition 3.22 of network identifiability of module G_{ji} is equivalently formulated as*

$$T(q,\theta_1) = T(q,\theta_0)\} \Rightarrow \{G_{ji}(q,\theta_1) = G_{ji}(q,\theta_0)\}. \tag{3.34}$$

Identifiability of every module holds for every model set that is globally network identifiable. However now the interesting question is whether the conditions can be relaxed, such that identifiability of a module is guaranteed, even when other modules are not identifiable.

In order to find identifiability conditions for a single module G_{ji}, assume without loss of generality that this module corresponds to the top row of $\check{T}_j$. Then define $\check{T}_{j,(i,\star)}$ as the top row of $\check{T}_j$, and $\check{T}_{j,(-i,\star)}$ by

$$\check{T}_j\left(T(q,\theta)\right) = \begin{bmatrix} \check{T}_{j,(i,\star)}\left(T(q,\theta)\right) \\ \check{T}_{j,(-i,\star)}\left(T(q,\theta)\right) \end{bmatrix} \tag{3.35}$$

So $\check{T}_{j,(-i,\star)}$ is $\check{T}_j$ with the row corresponding to node w_i removed. The following Theorem now specifies necessary and sufficient conditions for the identifiability condition (3.33).

Theorem 3.24. *Let $\mathcal{M}$ be a network model set as defined in Theorem 3.15, and let $\check{T}_j\left(T(q,\theta)\right)$ be defined by* (3.28)*,and $\check{T}_{j,(-i,\star)}\left(T(q,\theta)\right)$ by* (3.35)*, then:*

1. *Module G_{ji} of network model set $\mathcal{M}$ is globally network identifiable at $M(\theta_0)$ if and only if*

$$\text{rank}(\check{T}_j\left(T(q,\theta_0)\right)) > \text{rank}(\check{T}_{j,(-i,\star)}\left(T(q,\theta_0)\right)). \tag{3.36}$$

2. *Module G_{ji} of network model set $\mathcal{M}$ is globally network identifiable if and only if*

$$\text{rank}(\check{T}_j\left(T(q,\theta)\right)) > \text{rank}(\check{T}_{j,(-i,\star)}\left(T(q,\theta)\right)) \tag{3.37}$$

for all $\theta \in \Theta$.

Proof. Provided in Appendix 3.10.7. ■

The essential part of the theorem is that if the row of $\check{T}_j$ corresponding to node w_i is a linear independent row, then the module is identifiable. Note that there is no explicit requirement on the number of parameterized elements in Theorem 3.24. We do not require uniqueness of all modules, so we can have fewer equations than unknowns.

Example 3.25 (Example 3.21 continued)**.** *For node 4 there are three parameterized transfer functions, while there are only two external signals. To evaluate identifiability we use*

$$\check{T}_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & G_{32} \end{bmatrix}, \quad \check{T}_{4(-1,\star)} = \begin{bmatrix} 0 & 1 \\ 0 & G_{32} \end{bmatrix}. \tag{3.38}$$

In $\check{T}_4$ the first row is clearly linearly independent of the other rows, such that

$$\mathrm{rank}(\check{T}_{4(-1,\star)}) = 1 < \mathrm{rank}(\check{T}_4) = 2,$$

and the condition of Theorem 3.24 is satisfied for G_{41}. It can be shown that rows 2 and 3 of $\check{T}_4$ are linearly dependent, and so G_{42} and G_{43} are both not identifiable.

It should be noted that to guarantee identifiability of a single row or module it is likely not necessary to guarantee that implication (3.18) holds, as only a sub-matrix of T is used to determine the uniqueness of the row or module. The implications in (3.18), and (3.30) or (3.34) respectively may be written using a sub-matrix instead of T in order to obtain conditions that are both necessary and sufficient. In future research generalized conditions on feedthrough terms and instantaneous noise correlations locally around the node or module of interest may be formulated, instead of on the full network model.

3.7 Path-based identifiability conditions

3.7.1 Generic network identifiability

It is not intuitive to relate the rank conditions that have been formulated in Theorem 3.15, Corollary 3.20 and Theorem 3.24 to properties of the network. In this section the rank conditions that appear in the network identifiability results are formulated as topology based conditions. The core idea is that the rank of T depends on the topology of the network. We base our reasoning on concepts presented in Bazanella et al. (2017); Hendrickx et al. (2018), such as generic identifiability, which we adapt to our problem setting. Then the rank conditions for identifiability are adapted to this definition. The notion of *vertex disjoint paths* and its relation to the rank of the transfer matrix will be introduced, and used to formulate topological conditions under which the network is identifiable.

> **Definition 3.26** (Generic network identifiability)**.**
>
> - *$\mathcal{M}$ is generically network identifiable if for all models $M(\theta_1) \in \mathcal{M}$ the implication (3.19) holds for* almost all $M_0 \in \mathcal{M}$.
> - *Row j of network model set $\mathcal{M}$ is generically network identifiable if for all models $M(\theta_1) \in \mathcal{M}$ the implication (3.29) holds for* almost all $M_0 \in \mathcal{M}$.
> - *Module G_{ji} of network model set $\mathcal{M}$ is generically globally network identifiable for all models $M(\theta_1) \in \mathcal{M}$ the implication (3.33) holds for* almost all $M_0 \in \mathcal{M}$.

The only difference between Definitions 3.1 and 3.26 is the exception of a set of zero measure, i.e. identifiability of almost all models is accounted for. The consequences of this change in the definitions and an overview of the different identifiability notions are discussed in Section 3.7.4.

The rank conditions of Theorems 3.15 and 3.24, Corollary 3.20 can directly be formulated for the generic network identifiability.

Corollary 3.27. *The model set $\mathcal{M}$, row j of model set $\mathcal{M}$, or module G_{ji} of model set $\mathcal{M}$ is generically network identifiable by the conditions* 2) *of Theorem 3.15, Corollary 3.20, Theorem 3.24 respectively upon replacing the phrase "for all $\theta \in \Theta$" by "for* almost *all $\theta \in \Theta$".*

Proof. The proof is a trivial extension of the proof of Theorem 3.15 and the proof of Theorem 3.24. ■

3.7.2 Vertex disjoint paths

In van der Woude (1991), the name vertex is used for a node, and the rank of a transfer matrix is connected to the notion of a set of vertex disjoint paths. The notion of vertex disjoint paths can be used to formulate topological conditions under which a model set is generically network identifiable, following the approach in Bazanella et al. (2017). As defined in van der Woude (1991), two paths in a network between external signals or nodes are vertex disjoint if they have no common nodes, including their start and end nodes. For a set of l paths, these paths are vertex disjoint if every pair of paths is vertex disjoint.

The essential meaning is the following: If there exists a set of vertex disjoint paths from some excitations r_k, e_l to some nodes w_i, then every one of those nodes has 'its own' source of excitation. Note that when two paths are vertex disjoint, there may still exist modules that connect the nodes in the paths, and there may exist loops around the nodes.

The connection to the rank of a transfer matrix is on the basis of state-space systems in the following way. A parameterized state-space system is defined with matrices A, B, C, and the open-loop transfer from input to output is defined as $T_{ss} := C(sI - A)^{-1}B$. Then the *generic rank* of the transfer matrix T_{ss} is defined as the rank of T_{ss} for almost all values of parameters. This is formalized in the following theorem.

Theorem 3.28 (Theorem 2 from van der Woude (1991))**.** *Let G_Σ be the graph corresponding to the state-space system*

$$\dot{x} = Ax + Br, \quad w = Cx. \tag{3.39}$$

The maximum number of vertex disjoint paths in G_Σ from signals in r to signals in w equals the generic rank of T_{ss}.

3.7.3 Path-based identifiability conditions

The next step is to link the network of transfer functions represented by model M to a state-space formulation. In order to avoid difficulties with direct feedthrough terms we make the assumption that the modules G are strictly proper, and we can shift the H and R by a sample delay while also shifting the signals, i.e.

$$H(q)e(t) = H(q)q^{-1}e(t+1), \quad R(q)r(t) = R(q)q^{-1}r(t+1). \tag{3.40}$$

Shifting H and R by a delay does not have an influence on the network topology or the rank of the open-loop transfer function matrix $T(q)$.

The associated state-space system is defined in the following way. For each node $j = 1, \dots, L$ define a state-space system in observable canonical form with the state vector x_j and the state equation

$$\begin{aligned} x_j^+ &= A_j x_j + \sum_{i=1}^{L} B_{ji}^w w_i + \sum_{k=1}^{K} B_{jk}^r r_k + \sum_{l=1}^{p} B_{jl}^e e_l \\ w_j &= C_j x_j, \end{aligned} \tag{3.41}$$

where A_j and C_j have the structure

$$A_j = \begin{bmatrix} * & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ * & 0 & \ddots & 1 & 0 \\ * & 0 & \cdots & 0 & 1 \\ * & 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad C_j = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{3.42}$$

via the relations

$$\begin{aligned} G_{j\star} &= C_j(zI - A_j)^{-1}B_{j\star}^w \\ q^{-1}R_{j\star} &= C_j(zI - A_j)^{-1}B_{j\star}^r \\ q^{-1}H_{j\star} &= C_j(zI - A_j)^{-1}B_{j\star}^e. \end{aligned} \tag{3.43}$$

Note that $B_{ji}^w = 0$ if w_i is not an in-neighbor to w_j. The full network can be written in state-space form by interconnecting all of the individual state-space systems (3.41)

$$\begin{aligned} x^+ &= Ax + B^r r + B^e e, \\ w &= Cx, \end{aligned} \tag{3.44}$$

with

$$B^w = \begin{bmatrix} B_{11}^w & \cdots & B_{1L}^w \\ \vdots & \ddots & \vdots \\ B_{L1}^w & \cdots & B_{LL}^w \end{bmatrix}, \; B^r = \begin{bmatrix} B_{11}^r & \cdots & B_{1K}^r \\ \vdots & \ddots & \vdots \\ B_{L1}^r & \cdots & B_{LK}^r \end{bmatrix}, \; B^e = \begin{bmatrix} B_{11}^e & \cdots & B_{1p}^e \\ \vdots & \ddots & \vdots \\ B_{L1}^e & \cdots & B_{Lp}^e \end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_L \end{bmatrix}, \; C = diag(\{C_i\}_{i=1\cdots L}), \; A = diag(\{A_{ii}\}_{i=1\cdots L}) + B^w C. \tag{3.45}$$

The network topology of G is encoded by the block-structure of B^w. Associate the graph G_M with nodes w, r based on the topology of M, and associate graph G_Σ with nodes x, w, r based on the topology of (3.44). Graph G_Σ has some special structure, there always exists a path $x_{j\{n\}}, \ldots, x_{j\{1\}}$ due to the structure of A_j induced by the observable canonical form. There are some strong relations between the graphs G_M and G_Σ which are specified in the following lemma.

Lemma 3.29. *Let graph G_M be associated with nodes w, r based on the topology of M, and let graph G_Σ be associated with nodes x, w, r based on the topology of* (3.44)*, then:*

1. *In G_M node w_i is an in-neighbor of w_j if and only if in G_Σ there is a path $x_{i\{n\}}, x_{j\{s\}}, x_{j\{s-1\}}, \ldots, x_{j\{1\}}$ for some s;*
2. *In G_M node r_k is an in-neighbor of w_j if and only if in G_Σ there is a path $r_k, x_{j\{s\}}, x_{j\{s-1\}}, \ldots, x_{j\{1\}}$ for some s;*
3. *In G_M node e_l is an in-neighbor of w_j if and only if in G_Σ there is a path $e_l, x_{j\{s\}}, x_{j\{s-1\}}, \ldots, x_{j\{1\}}$ for some s;*
4. *For some sequence $k_1, \ldots, k_n$ the path $r_k, w_{k_1}, \ldots, w_{k_n}, w_j$ exists in G_M if and only if the path $r_k, x_{k_1}, \ldots, x_{k_n}, x_j, w_j$ exists[a] in G_Σ;*
5. *For some sequence $k_1, \ldots, k_n$ the path $e_l, w_{k_1}, \ldots, w_{k_n}, w_j$ exists in G_M if and only if the path $e_l, x_{k_1}, \ldots, x_{k_n}, x_j, w_j$ exists[a] in G_Σ.*

[a] Technically every x_k is to be replaced with $x_{k\{s_k\}}, \cdots, x_{k\{1\}}$ for some s_k

Proof. Provided in Appendix 3.10.8. ■

The interpretation of the lemma is as follows. When w_i is an in-neighbor of w_j in the transfer function network, then in the state-space network this is represented as a path of nodes that runs only through the states in x_j. A path that passes through nodes $w_{k_1}, \ldots, w_{k_n}$ in the transfer function network is associated with the path that passes through states in $x_{k_1}, \ldots, x_{k_n}$ in the state-space network. This implies that paths in G_M and their associated paths in G_Σ share their vertex joint/disjoint properties.

Proposition 3.30. *Let graph G_M be associated with nodes w, r based on the topology of M, and let graph G_Σ be associated with nodes x, w, r based on the topology of* (3.44)*. The paths $r_k, w_{k_1}, \ldots, w_{k_n}, w_j$ and $r_l, w_{l_1}, \ldots, w_{l_n}, w_i$ in G_M are disjoint if and only if the paths $r_k, x_{k_1}, \ldots, x_{k_n}, x_j, w_j$ and $r_l, x_{l_1}, \ldots, x_{l_n}, x_i, w_i$ in G_Σ are disjoint.*

Proof. Follows directly from Lemma 3.29. ■

Now the maximum number of disjoint paths can be linked to the generic rank.

Proposition 3.31. *Let $\mathcal{U}$ be a set of white noises e_l and external variables r_k, and let $\mathcal{Y}$ be a set of nodes w_j. Define the set of open-loop transfer functions from excitations in $\mathcal{U}$ to nodes in $\mathcal{Y}$ that is generated by the model set as*

$$\mathcal{T} := \{T_{\mathcal{Y}\mathcal{U}}(q,\theta) \mid \theta \in \Theta\}. \tag{3.46}$$

The maximum number of vertex disjoint paths in G_M from excitations in $\mathcal{U}$ to nodes in $\mathcal{Y}$ is equal to the generic rank of $T_{\mathcal{Y}\mathcal{U}} \in \mathcal{T}$.

Proof. Provided in Appendix 3.10.9. ■

The rank conditions on $\check{T}_j$ of Corollary 3.27 can now be evaluated by checking whether there are a sufficient number of vertex disjoint paths from selected external signals to node signals.

Proposition 3.32. *Let $\mathcal{M}$ be a set of network models M with strictly proper modules in G. Let $\mathcal{Y}_j$ be the set of nodes w_k which are an input to a parameterized $G_{jk}(\theta)$, and let α_j be the the cardinality of $\mathcal{Y}_j$. Let $\mathcal{U}_j$ be the set of external signals r_k, e_l that are an input to* non-*parameterized R_{jk}, H_{jl}.*

1. *The model set $\mathcal{M}$ is generically network identifiable if and only if condition i) of Theorem 3.15 holds and for each j, there is a set of α_j vertex disjoint paths from excitations in $\mathcal{U}_j$ to nodes in $\mathcal{Y}_j$.*
2. *Row j of model set $\mathcal{M}$ is generically network identifiable if and only if condition i) of Corollary 3.20 holds and there is a set of α_j vertex disjoint paths from excitations in $\mathcal{U}_j$ to nodes in $\mathcal{Y}_j$.*
3. *For module G_{ji}, let $\bar{\mathcal{Y}}_j = \mathcal{Y}_j \setminus w_i$. Module G_{ji} of model set $\mathcal{M}$ is generically network identifiable if and only if there exists a set $\mathcal{P}$ of the maximum number of vertex disjoint paths from signals in $\mathcal{U}_j$ to nodes in $\bar{\mathcal{Y}}_j$, and there is an additional path from signals in $\mathcal{U}_j$ to w_i, such that this path and the paths in $\mathcal{P}$ are vertex disjoint.*

Proof. Obtained by combining Corollary 3.27 with Proposition 3.31. ■

In order to satisfy condition 1) or 2) there is an implicit requirement on the number of available external signals, which is directly related to the maximum number of parameterized elements in conditions 1) and 2) of Corollary 3.27. For condition 3) there is no minimum number of external signals, but there is the implicit requirement that there is a 'surplus' excitation that can form a vertex disjoint path to the module of interest.

In order to check the conditions of Proposition 3.32, all that must be done is check which transfer functions are parameterized, and check whether the necessary paths are present in the network. This is illustrated in an example.

Example 3.33 (Example 3.25 continued)**.** *Topology based conditions for identifiability are checked for various modules in the network in Figure 3.5. When checking identifiability of modules that map into node w_4 we see that the in-neighbors are*

$\mathcal{Y}_4 = \{w_1, w_2, w_3\}$, so there are $\alpha_4 = 3$ parameterized modules. There are only 2 excitations, which can never form 3 vertex disjoint paths, so the row is not generically network identifiable.

For identifiability of module G_{42} there are two vertex disjoint paths from external signals to the other inputs w_1, w_3, and there is no surplus excitation available for w_2. However for identifiability of module G_{41} there is just one vertex disjoint path from external signals to the other inputs w_2, w_3, and there is the surplus excitation available for w_1, so G_{41} is generically network identifiable.

3.7.4 Discussion on definition of identifiability

Path-based conditions are based on generic rank, and not 'standard' rank. The difference between the two definitions of identifiability is the exclusion of a zero-measure set of models, so network identifiability is stricter than generic network identifiability. When one model in $\mathcal{M}$ is not identifiable, then $\mathcal{M}$ is not network identifiable, but it can be generically network identifiable. Next an example is given of a single non-identifiable model in the model set, which is in particular relevant in case the objective is to identify the network topology.

Example 3.34. *Suppose we have a parameterized set of models as depicted in Figure 3.6, with*

$$G = \begin{bmatrix} 0 & G_{12}(\theta) \\ G_{21}(\theta) & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad T^0 = \begin{bmatrix} \frac{1}{1-G_{12}^0 G_{21}^0} \\ \frac{G_{21}^0}{1-G_{12}^0 G_{21}^0} \end{bmatrix}.$$

Identifiability of G_{12} and G_{21} is determined from the rank of $\check{T}_1 = \frac{G_{21}}{1-G_{12}G_{21}}$ and $\check{T}_2 = \frac{1}{1-G_{12}G_{21}}$ respectively. For all θ where $G_{21}(\theta) = 0$ the $\check{T}_1$ loses rank and G_{12} is not identifiable.

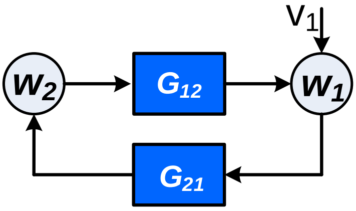


Figure 3.6: A closed-loop network representing a set of models.

In a situation where the topology is known, i.e. it is known that $G_{21} \neq 0$, then we want to classify the model set as identifiable, which can be done with generic network identifiability. However when the topology is not known a-priori, we would like to determine whether G_{12} and G_{21} are zero or non-zero. Then the possibility that $G_{21}^0 = 0$ must be taken into account such that the generic network identifiability concept is less attractive, and the global network identifiability concept is more powerful.

3.8 Signal ordering

In Assumption 2.2 we have formulated a condition on an ordering property of the signals. It should be determined whether Assumption 2.2 is reasonable. To that end in this section we will further discuss this assumption and how it can be dealt with. We will investigate whether the information on the ordering of signals can be extracted from data.

Our definitions of models and model sets in Section 2.1 only consider models that have the ordering property, and a particular rank in the noise model. So, for discussing the situation of models that do not have this property, we need to slightly adapt Definition 2.6.

Definition 3.35 (Network model without ordering)**.** *A network model without ordering property is defined by the quadruple*

$$M = (G, R, \tilde{H}, \tilde{\Lambda}) \tag{3.47}$$

with $\tilde{H} \in \mathbb{R}^{L\times L}(z)$ monic, $\tilde{\Lambda} \in \mathbb{R}^{L\times L}$ with any rank $p \leq L$, and G and R as defined before in Definition 2.6.

First of all, if we are considering network identifiability at a particular known unordered model $M_0 = (G_0, R_0, \tilde{H}_0, \tilde{\Lambda}_0)$, then the covariance matrix $\tilde{\Lambda}_0$ carries the information of the rank p as well as the information for re-ordering the node signals w in such a way that, after reordering, the model satisfies the ordering property of Assumption 2.2. This can be understood by realizing that rank $\tilde{\Lambda}_0 = p$, and that there exists a permutation matrix Π such that $[I_p \;\, 0]\Pi^T\tilde{\Lambda}_0\Pi[I_p \;\, 0]^T = \Lambda_0$, the rank-$p$ covariance matrix of the ordered model. That same permutation matrix can then be applied to w, to reorder the node signals in the model so as to arrive at its ordered equivalent. So when considering a particular known model, the model information intrinsically contains the information how to order the signals to satisfy the ordering property.

For the mode general situation where M_0 is unknown, the required information for determining p and for reordering the node signals can be retrieved from data, so from T_{wr} and $\Phi_{\bar{v}}(\omega)$. In particular we can observe that on the basis of

$$\bar{v} = (I - G)^{-1}v$$

and $(I - G)$ being full rank, it is clear that rank $\Phi_{\bar{v}} =$ rank $\Phi_v = p$, and more specifically, by using the monicity property of $\tilde{H}$, that rank $\Phi_{\bar{v}}^{\infty} =$ rank $\tilde{\Lambda} = p$. So for an unknown model $M(\theta_0)$, p can be obtained directly from $\Phi_{\bar{v}}^{\infty}(\theta_0)$. A similar situation occurs for the ordering of signals as formulated in Assumption 2.2, as is formulated next.

Proposition 3.36. *Consider a network model $M_0 = M(\theta_0)$ according to Definition 3.35, with rank $\Phi_{\bar{v}}(\theta_0) = p$. If either one of the following conditions is satisfied:*

1. $G^\infty(\theta_0) = 0$;

2. $G^\infty(\theta_0)$ has a known pattern of 0's, that guarantees that there are no algebraic loops, and $\Phi_{\bar{v}}^\infty(\theta_0)$ is diagonal;

3. Each row of $[G^\infty(\theta_0)\ R^\infty(\theta_0)]$ has at most K nonzero elements, and for each $i \in \mathcal{N}$, the matrix $\check{T}_i^\infty(\theta_0)$ (3.24) has full row rank,

then on the basis of $T_{wr}^\infty(\theta_0)$ and $\Phi_{\bar{v}}^\infty(\theta_0)$ a permutation matrix Π can be constructed that reorders the node signals w in such a way that the permuted model satisfies the ordering property as meant in Assumption 2.2.

Proof. Provided in Appendix 3.10.6. ■

The reasoning that underlies this result, is that under the formulated conditions the covariance matrix $\tilde{\Lambda}_0$ can be uniquely retrieved from the data. And based on $\tilde{\Lambda}_0$ a permutation matrix can then be found that reorders the node signals into a (reordered) model that satisfies the ordering property.
The conditions of this Proposition are basically the same as the ones applied in Propositions 3.7, 3.8 and 3.9 for analyzing identifiability.

The results in this section show that the ordering property of Assumption 1 is not a restriction if we consider the identifiability of a model set at a particular model. This is due to the fact that in that particular model, either the model information or the measurement data in the form of T_{wr}^∞ and $\Phi_{\bar{v}}^\infty$ carry enough information to find a permutation matrix to arrive at a permuted model that does satisfy Assumption 1.

3.9 Conclusions

The concept of network identifiability has been introduced as a way to describe whether network models can be distinghuished from each other. Conditions on the modeled topology, noise correlations and presence of external excitations can be formulated and checked explicitly such that global network identifiability of a dynamic network model set or module is guaranteed. Conditions that are based on paths present in the network have been formulated such that generic identifiability can rather easily be checked. It should be noted that the path-based conditions only apply to the situation that a set of zero-measure models is excluded from the model set.

Having an independent excitation available at every node is sufficient to guarantee network identifiability. When a-priori knowledge on a network is encoded into the model set, then the requirements on the presence of excitation is reduced. Identifiability conditions have been formulated for network models that includes noises that are possibly correlated and rank-reduced, and networks that may contain algebraic loops. With these reduced requirements on the network model we can describe more practical situations, for example situations where noises are correlated. When a network identifiable model set is combined with informative data, then unique network models can be retrieved from the data set. The identifiability results may be used to prove consistent estimation of all modules in a network. Other applications of the identifiability results are for example to identify the network topology, or to detect where the direct feedthrough terms of a network are.

3.10 Appendix

3.10.1 Proof of Proposition 3.7

Since in the considered situation

$$T_{we}(\theta) := (I - G(\theta))^{-1}H(\theta)$$

has an upper $p \times p$ part which is monic, while

$$\Phi_{\bar{v}}(\theta) = T_{we}(\theta)\Lambda(\theta)T_{we}(\theta)^* \tag{3.48}$$

it follows that (3.48) satisfies the conditions of the unique spectral factorization in Lemma 2.3a, if $p < L$. If $p = L$ it satisfies the conditions of the standard spectral factorization. Therefore T_{we} and Λ are uniquely determined by $\Phi_{\bar{v}}$, or in other words

$$\{\Phi_{\bar{v}}(\theta_1) = \Phi_{\bar{v}}(\theta_0)\} \Longrightarrow \left\{ \begin{array}{c} T_{we}(\theta_1) = T_{we}(\theta_0) \\ \Lambda(\theta_1) = \Lambda(\theta_0) \end{array} \right. .$$

Since $T_{wr}(\theta_1) = T_{wr}(\theta_0)$ is in the premise of (3.7) and $\Lambda(\theta_1) = \Lambda(\theta_0)$ is implied by the premise of the equality of the spectra, as indicated above, the result follows directly. ■

3.10.2 Proof of Proposition 3.8

First we treat the full-rank situation that $p = L$.

In this situation

$$\Phi_{\bar{v}}(z, \theta) := (I - G(\theta))^{-1}H(\theta)\Lambda(\theta)H(\theta)^*(I - G(\theta))^{-*}$$

and using the property that H is monic leads to

$$\Phi_{\bar{v}}^{\infty}(\theta) := \lim_{z \to \infty} \Phi_{\bar{v}}(z, \theta) = (I - G^{\infty}(\theta))^{-1}\Lambda(\theta)(I - G^{\infty}(\theta))^{-T}.$$

The algebraic loop condition now implies that $\Pi^T(I - G^{\infty}(\theta))^{-1}\Pi$ is upper unitriangular[2] and (leaving out arguments θ for brevity):

$$\Pi^T\Phi_{\bar{v}}^{\infty}\Pi = \underbrace{\Pi^T(I - G^{\infty})^{-1}\Pi}_{L} \cdot \underbrace{\Pi^T\Lambda\Pi}_{D} \cdot \underbrace{\Pi^T(I - G^{\infty})^{-T}\Pi}_{L^T}.$$

[2] upper unitriangular is upper triangular with 1's on the diagonal.

With D being diagonal and L upper unitriangular, this represents a unique LDL^T decomposition of the permuted spectrum. As a result Λ is uniquely determined from $\Phi_{\bar{v}}$.

Spectral factorization of $\Phi_{\bar{v}}$ leads to a unique decomposition

$$\Phi_{\bar{v}} = \hat{H}\hat{\Lambda}\hat{H}^*$$

with $\hat{H}$ monic, stable and minimum-phase, but $\hat{\Lambda}$ not necessarily diagonal. Since $\hat{\Lambda}$ is full rank, there is a nonsingular matrix B such that $\hat{\Lambda} = B\Lambda B^T$, leading to the unique spectral decomposition:

$$\Phi_{\bar{v}} = \hat{H}B\Lambda B^T\hat{H}^*,$$

where $\hat{H}B = T_{we}$. As a result, T_{we} is uniquely determined from $\Phi_{\bar{v}}$, and the proof follows along the same steps as in the proof of Proposition 3.7.

Now we turn to the situation $p < L$.

When applying the spectral decomposition of Lemma 2.3b to Φ_v it follows that

$$\Phi_{\bar{v}}(z,\theta) = (I - G(\theta))^{-1}\breve{H}(\theta)\breve{\Lambda}(\theta)\breve{H}(\theta)^*(I - G(\theta))^{-*}$$

with $\breve{H}$ square and monic, and structured according to

$$\breve{H} = \begin{bmatrix} H_a & 0 \\ H_b - \Gamma & I \end{bmatrix}, \text{ and } \breve{\Lambda} = \begin{bmatrix} I \\ \Gamma \end{bmatrix} \Lambda \begin{bmatrix} I \\ \Gamma \end{bmatrix}^T.$$

Since by assumption Φ_v^∞ is diagonal, it follows that $\Gamma := \lim_{z\to\infty} H_b(z) = 0$ and

$$\breve{\Lambda} = \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix}.$$

As a result

$$\Phi_{\bar{v}}^\infty = (I - G^\infty(\theta))^{-1}\breve{\Lambda}(\theta)(I - G^\infty(\theta))^{-T}$$

with $\breve{\Lambda}(\theta)$ diagonal. Then exactly the same reasoning as above with a permutation of the signals to turn $(I - G^\infty)^{-1}$ into a unitriangular matrix, shows that $\breve{\Lambda}$ and therefore also Λ is uniquely determined from $\Phi_{\bar{v}}$.

With Λ known, the decomposition $\Phi_{\bar{v}} = T_{we}\Lambda T_{we}^*$ uniquely determines T_{we} from $\Phi_{\bar{v}}$. The proof then follows the same same steps as in the proof of Proposition 3.7. ■

3.10.3 Proof of Proposition 3.9

This proof consists of 2 steps. The first step is to use T_{wr} to uniquely determine the feedthrough of G, i.e.

$$T_{wr}^\infty(\theta_1) = T_{wr}^\infty(\theta_0) \Rightarrow G^\infty(\theta_1) = G^\infty(\theta_0). \tag{3.49}$$

The left hand side of the above implication can be written as

$$(I - G^\infty(\theta_0))T_{wr}^\infty(\theta_1) = R^\infty(\theta_0). \tag{3.50}$$

Consider row i of the matrix equation (3.50), and apply the following reasoning for each row separately. By inserting the permutation matrices P_i and Q_i, defined in (3.22)-(3.23), we obtain for row i:

$$(I - G^{\infty}(\theta_0))P_i^{-1}P_iT_{wr}^{\infty}(\theta_1)Q_i = R^{\infty}(\theta_0)Q_i \tag{3.51}$$

leading to

$$\begin{aligned}(I - G^{\infty}(\theta_0))_{i\star}^{(1)}T_i^{(1)}(\theta_1) + (I - G^{\infty})_{i\star}^{(2)}T_i^{(2)}(\theta_1) = \\ = \begin{bmatrix} R^{\infty}{}_{i\star}^{(1)} & R^{\infty}{}_{i\star}^{(2)}(\theta_0)\end{bmatrix},\end{aligned} \tag{3.52}$$

with $P_iT_{wr}^{\infty}(\theta_1)Q_i = \begin{bmatrix} T_i^{(1)}(\theta_1) \\ T_i^{(2)}(\theta_1)\end{bmatrix}$. Note that $\check{T}_i^{\infty}(\theta) = T_i^{(1)}(\theta)\begin{bmatrix} I_{K-\beta_i} \\ 0 \end{bmatrix}$ as defined by (3.24).

When considering the left $1 \times (K - \beta_i)$ block of the vector equation (3.52), while using the expression for $\check{T}_i^{\infty}(\theta)$ above, we can write

$$(I - G^{\infty}(\theta_0))_{i\star}^{(1)}\check{T}_i^{\infty}(\theta_1) + \rho(\theta_1) = R^{\infty}{}_{i\star}^{(1)}, \tag{3.53}$$

with $\rho(\theta_1)$ the left $1 \times (K - \beta_i)$ block of $(I - G^{\infty})_{i\star}^{(2)}T_i^{(2)}(\theta_1)$. Now $\rho(\theta_1)$ and $R^{\infty}{}_{i\star}^{(1)}$ are independent of parameter θ_0, which implies that, if $\check{T}_i^{\infty}(\theta_1)$ has full row rank, then all the parametrized elements in $(I - G^{\infty}(\theta_0))_{i\star}$ are uniquely determined.

Then the second step is to determine Λ and T_{we}. By writing the spectrum of $\bar{v}$ as

$$\Phi_{\bar{v}}^{\infty} = (I - G^{\infty})^{-1}H^{\infty}(\theta)\Lambda(\theta)(H^{\infty}(\theta))^T(I - G^{\infty})^{-T}$$

we obtain through pre- and post-multiplication:

$$(I - G^{\infty})\Phi_{\bar{v}}^{\infty}(I - G^{\infty})^T = \begin{bmatrix} \Lambda(\theta) & \Lambda(\theta)\Gamma^T(\theta) \\ \Gamma(\theta)\Lambda(\theta) & \Gamma(\theta)\Lambda(\theta)\Gamma^T(\theta)\end{bmatrix}$$

where $\Gamma := \lim_{z\to\infty} H_b(z,\theta)$. For given G^{∞} (from step 1), and given $\Phi_{\bar{v}}$, this equation provides a unique Λ, such that T_{we} can be uniquely obtained from

$$\Phi_{\bar{v}} = T_{we}(\theta)\Lambda T_{we}^{*}(\theta). \tag{3.54}$$

The proof then follows the same steps as the proof of Proposition 1. ■

3.10.4 Proof of Theorem 3.10

a) It will be shown that under the condition of the theorem, the equality $T(q,\theta) = T(q,\theta_0)$ implies $M(\theta) = M(\theta_0)$ for all $\theta \in \Theta$. With the definition of Θ_0, the equality of the T-matrices implies that we can restrict to $\theta \in \Theta_0$. That same equality induces

$$(I - G(\theta))^{-1}U(\theta) = (I - G(\theta_0))^{-1}U(\theta_0) \tag{3.55}$$

and postmultiplication with Q leads to

$$(I-G(\theta))^{-1}\begin{bmatrix}D(\theta) & F(\theta)\end{bmatrix}=(I-G(\theta_0))^{-1}\begin{bmatrix}D(\theta_0) & F(\theta_0)\end{bmatrix},$$

with $D(\theta)$ diagonal and full rank for all $\theta \in \Theta_0$.
The left square $L \times L$ blocks in both sides of the equation can now be inverted to deliver $D(\theta)^{-1}(I - G(\theta)) = D(\theta_0)^{-1}(I - G(\theta_0))$. Due to zeros on the diagonal of $G(\theta)$ and $G(\theta_0)$ and the diagonal structure of $D(\theta)$ and $D(\theta_0)$, it follows that $D(\theta) = D(\theta_0)$ and consequently $G(\theta) = G(\theta_0)$. Then by (3.55) it follows that $U(\theta) = U(\theta_0)$ and $M(\theta) = M(\theta_0)$.
b) For part (b) it needs to be shown that the implication under (a) holds true for *any* $M(\theta_0)$ in $\mathcal{M}$. It is direct that this is true, following a similar reasoning as above, if we extend the parameter set to be considered from Θ_0 to Θ. ■

3.10.5 Proof of Theorem 3.15

We will first provide the proof for situation (1).
The left hand side of the implication (3.19) can be written as

$$(I - G(\theta))T = U(\theta), \tag{3.56}$$

where we use shorthand notation $T = T(\theta_0)$, $G(\theta) = G(\theta_1)$ and $U(\theta) = U(\theta_1)$. Consider row i of the matrix equation (3.56), and apply the following reasoning for each row separately. By inserting the permutation matrices P_i and Q_i, defined in (3.26),(3.27) we obtain for row i:

$$(I - G(\theta))_{i\star}P_iP_i^{-1}TQ_i = U_{i\star}(\theta)Q_i \tag{3.57}$$

leading to

$$(I - G(\theta))_{i\star}^{(1)}T_i^{(1)} + (I - G)_{i\star}^{(2)}T_i^{(2)} = \begin{bmatrix}U_{i\star}^{(1)} & U(\theta)_{i\star}^{(2)}\end{bmatrix}, \tag{3.58}$$

with $P_i^{-1}TQ_i = \begin{bmatrix}T_i^{(1)}\\ T_i^{(2)}\end{bmatrix}$. Note that $\check{T}_i = T_i^{(1)}\begin{bmatrix}I_{K+p-\beta}\\ 0\end{bmatrix}$.

Sufficiency:
When considering the left $1 \times (K + p - \beta_i)$ block of the vector equation (3.58), while using the expression for $\check{T}_i$ above, we can write

$$(I - G(\theta))_{1\star}^{(1)}\check{T}_i + \rho = U_{i\star}^{(1)}, \tag{3.59}$$

with ρ the left $1 \times (K + p - \beta_i)$ block of $(I - G)_{i\star}^{(2)}T_i^{(2)}$.
Now ρ and $U_{i\star}^{(1)}$ are independent of θ, which implies that, if $\check{T}_i$ has full row rank, then all the parametrized elements in $(I - G(\theta))_{i\star}$ are uniquely determined. Then through (3.58) the parametrized elements in $U_{i\star}(\theta)$ are also uniquely determined.

By assumption we know that one solution to (3.56) is given by $G(\theta_0)$ and $U(\theta_0)$. Since the solution is unique, and $G(\theta_0)$ and $U(\theta_0)$ are a possible solution we know that $G(\theta_0)$ and $U(\theta_0)$ must be the only solution. This proves the validity of the implication (3.19).

Necessity of condition 2:
If the matrix $\check{T}_i(\theta_0)$ is not full row rank, then it has a non-trivial left nullspace. Let the rational transfer matrix $X \neq 0$ of dimension $1 \times \alpha_i$ be in the left nullspace of $\check{T}_i$. Then there also exists a proper, rational and stable X_p in the left nullspace of $\check{T}_i$. Then (3.59) can also be written as

$$\left((I - G(\theta))^{(1)}_{i\star} + X_p\right)\check{T}_i + \rho = U^{(1)}_{i\star}. \tag{3.60}$$

By the formulated assumptions (a) and (b) it holds that each parameterized transfer function can be any proper rational transfer function, and that these parameterized transfer functions do not share any parameters. This implies that $G(\theta_1)_{i\star} \in \mathcal{M}$ and $(G(\theta_1)_{i\star} - X_p) \in \mathcal{M}$ refer to two different model rows of G in the model set, that generate the same network transfer function T. Hence implication (3.19) can not hold.

Necessity of condition 1:
If $\alpha_i + \beta_i > K + p$, then $\check{T}_i(\theta_0)$ will be a tall matrix which can never have a full row rank. Then because of the necessity of the row rank condition on $\check{T}_i(\theta_0)$, necessity of condition 1 follows immediately.

Proof of situation (2): For all $\theta \in \Theta$:
For every $\theta \in \Theta$ we can construct $T(\theta)$ with related $\check{T}_i(\theta)$ of full row rank, and the reasoning as presented before fully applies. If for some $\theta \in \Theta$ we can not construct this full row rank $\check{T}_i(\theta)$ there exists a model in the model set which is not identifiable, and hence the model set is not globally network identifiable in $\mathcal{M}$. ■

3.10.6 Proof of Proposition 3.36

The expression for $\Phi_{\bar{v}}$ is given by (discarding arguments θ_0):

$$\Phi_{\bar{v}} = [I - G]^{-1}\tilde{H}\tilde{\Lambda}\tilde{H}^*[I - G]^{-*}. \tag{3.61}$$

while $T_{wr} = [I - G]^{-1}R$. Because $\tilde{H}$ is monic, the expression for $\Phi^{\infty}_{\bar{v}}$ reduces to:

$$\Phi^{\infty}_{\bar{v}} = [I - G^{\infty}]^{-1}\tilde{\Lambda}[I - G]^{-*}. \tag{3.62}$$

We are now going to show that under the different conditions, $\tilde{\Lambda}$ can be uniquely derived from $\Phi^{\infty}_{\bar{v}}$ and T^{∞}_{wr}.

Situation of strictly proper modules (Proposition 3.7).
Since we know that $G^{\infty} = 0$ it follows immediately from (3.62) that $\Phi^{\infty}_{\bar{v}} = \tilde{\Lambda}$, showing that $\tilde{\Lambda}$ can be directly obtained from $\Phi^{\infty}_{\bar{v}}$.

Situation of diagonal Λ and no algebraic loops (Proposition 3.8).
If Φ^{∞}_{v} is diagonal then also $\tilde{\Lambda}$ is diagonal. We consider (3.62). Based on the algebraic loop condition, we can construct a permutation matrix Π such that $\Pi^T(I - G^{\infty})^{-1}\Pi$ is upper unitriangular. Then:

$$\Pi^T\Phi^{\infty}_{\bar{v}}\Pi = \\ \underbrace{\Pi^T(I - G^{\infty})^{-1}\Pi}_{L} \cdot \underbrace{\Pi^T\tilde{\Lambda}\Pi}_{D} \cdot \underbrace{\Pi^T(I - G^{\infty})^{-T}\Pi}_{L^T}.$$

With D being diagonal and L upper unitriangular, this represents a unique LDL^T decomposition of the permuted spectrum. As a result $\tilde{\Lambda}$ is uniquely determined from $\Phi_{\bar{v}}^{\infty}$.

Situation of algebraic loops (Proposition 3.9).
The proof of Proposition 3.9 shows that under the given conditions, G^{∞} is uniquely determined from T_{wr}. Then (3.62) leads to the expression

$$[I - G^{\infty}]\Phi_{\bar{v}}^{\infty}[I - G^{\infty}]^* = \tilde{\Lambda}. \tag{3.63}$$

showing that $\tilde{\Lambda}$ can be uniquely determined.

In all three situations considered, the matrix $\tilde{\Lambda}$ is uniquely determined from $\Phi_{\bar{v}}^{\infty}$ and possibly T_{wr}^{∞}. Then there exists a permutation matrix Π that reorders the signals v in such a way that $\Pi\tilde{\Lambda}\Pi^T$ is a matrix of which the left upper $p \times p$ part is full rank. If we apply this reordering of signals, determined by Π, to the node signals w, then we arrive at a permuted model that has the ordering property, according to Assumption 2.2. ∎

3.10.7 Proof of Theorem 3.24

The left hand side of the implication (3.33) can be written as

$$(I - G(\theta))T = U(\theta), \tag{3.64}$$

where we use shorthand notation $T = T(\theta_0)$, $G(\theta) = G(\theta_1)$ and $U(\theta) = U(\theta_1)$. By inserting the permutation matrices P and Q as in (3.28) we obtain for row j:

$$(I - G(\theta))_{j\star}PP^{-1}TQ = U_{j\star}(\theta)Q \tag{3.65}$$

leading to

$$(I - G(\theta))_{j\star}^{(1)}T_j^{(1)} + (I - G)_{j\star}^{(2)}T_j^{(2)} = \begin{bmatrix} U_{j\star}^{(1)} & U(\theta)_{j\star}^{(2)} \end{bmatrix}, \tag{3.66}$$

with $P^{-1}TQ = \begin{bmatrix} T_j^{(1)} \\ T_j^{(2)} \end{bmatrix}$. Note that $\check{T}_j = T_j^{(1)} \begin{bmatrix} I_{K+p-\beta} \\ 0 \end{bmatrix}$. The right-hand block in (3.66) corresponding to $U(\theta)_{j\star}^{(2)}$ does not add to the uniqueness of the module of interest since it is fully parameterized (conditions *a,b* of Theorem 3.15), so equivalently we can consider

$$(I - G(\theta))_{j\star}^{(1)}\check{T}_j + \rho = U_{j\star}^{(1)}, \tag{3.67}$$

with ρ the left $1 \times (K + p - \beta)$ block of $(I - G)_{j\star}^{(2)}T_j^{(2)}$. Now since ρ and $U_{j\star}^{(1)}$ are independent of θ we have that $(I - G(\theta))_{ji}^{(1)}$ is uniquely specified if and only if $(I - G(\theta))_{ji}^{(1)}$ is uniquely specified in the left-nullspace of $\check{T}_j$.

Sufficiency:
Define some transfer matrix $X(q)$ of dimension $(K + p - \beta) \times 1$ with the following properties:

- $\check{T}_{j(-i,\star)}(q, \theta_0)X(q) = 0$, and

- $\check{T}_{j(i,\star)}(q,\theta_0)X(q) \neq 0$,

where $\check{T}_{j(-i,\star)}$ and $\check{T}_{j(i,\star)}$ are defined in (3.35). This X exists because condition (3.36) requires that $\check{T}_{j(-i,\star)}(q,\theta_0)$ is not full column rank, and condition (3.36) implies that $\check{T}_{j(i,\star)}(q,\theta_0)$ is linearly independent from the rows of $\check{T}_{j(-i,\star)}(q,\theta_0)$. Now define an $(K+p-\beta)\times(K+p-\beta)$ full rank transfer matrix $Z(q)$ which has X as its first column. Then (3.67) can be post-multiplied with Z to obtain an equivalent set of equations, leaving the set of solutions for G_{ji} invariant. The first column of $\check{T}_jZ$ is

$$\check{T}_j(q,\theta_0)X(q) = \begin{bmatrix} \check{T}_{j(i,\star)}(q,\theta_0)X(q) \\ 0 \end{bmatrix}, \tag{3.68}$$

such that, for this choice of Z, G_{ji} can be uniquely determined from

$$(I-G(\theta))^{(1)}_{j\star} \begin{bmatrix} \check{T}_{j(i,\star)}X \\ 0 \end{bmatrix} = (U^{(1)}_{j\star} - \rho)X. \tag{3.69}$$

If G_{ji} is unique for this particular choice of Z, it must be unique in the original problem also.

Necessity:
The converse of condition (3.36) is that $\operatorname{rank}\big(\check{T}_j(q,\theta_0)\big) = \operatorname{rank}\big(\check{T}_{j(-i,\star)}(q,\theta_0)\big)$. In this case the row of $\check{T}_j(q,\theta_0)$ corresponding to $G_{ji}(\theta)$ is linearly dependent on other rows of $\check{T}_j(q,\theta_0)$. When $\check{T}_{j(i,\star)}$ is linearly dependent on another row $\check{T}_{j(k,\star)}$, an equation equivalent to (3.67) can be created where the element G_{j1} and row $\check{T}_{j(i,\star)}$ are deleted, and where $(G_{ji}F+G_{jk})$ replaces G_{jk}, such that G_{ji} can not uniquely be distinguished.

Proof of situation (2): For all $\theta \in \Theta$:
For every $\theta \in \Theta$ we can construct $T(\theta)$ with related $\check{T}_j(\theta)$. If condition (3.36) applies for every model as stated by condition (3.37), then the reasoning as presented before fully applies to every model. If for some $\theta \in \Theta$ the condition (3.36) is not met, there exists a model in the model set which is not identifiable, and hence the model set is not globally network identifiable in $\mathcal{M}$. ■

3.10.8 Proof of Lemma 3.29

Proof of statement 1: The when w_i is an in-neighbor of w_j this implies $G_{ji} \neq 0$, which implies $B^w_{ji} \neq 0$, such that the ji block of A denoted with $A_{\{ji\}} = B^w_{ji}C_i$ has a non-zero first column since $C_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, such that $x_{i,\{1\}}$ is an in-neighbor of $x_{j,\{s\}}$ for some s. The implications act also in the other direction. Upon replacing w_i with r_k or e_l, then it is implied that $R_{jk} \neq 0$ or $H_{jl} \neq 0$ respectively, such that $B^r_{jk} \neq 0$ or $B^e_{jl} \neq 0$ respectively, which implies that r_k or e_l is an in-neighbor of $x_{j,\{s\}}$ for some s.

Statement 2 is a direct consequence of statement 1. ■

3.10.9 Proof of Proposition 3.31

By Proposition 3.30 the number of vertex disjoint paths in G_M from excitations in $\mathcal{U}$ to nodes in $\mathcal{Y}$ is equal to number of vertex disjoint paths in G_Σ from excitations in $\mathcal{U}$ to nodes in $\mathcal{Y}$ for every model in $\mathcal{M}$, such that the maximum number of vertex disjoint paths over the set of models is equal.

For each model θ in Θ the open-loop transfer function is

$$T_{\mathcal{Y}\mathcal{U}}(z,\theta) = C_{\mathcal{Y}}(zI - A(\theta))^{-1}B_{\mathcal{U}}(\theta), \tag{3.70}$$

where $C_{\mathcal{Y}}$ and $B_{\mathcal{U}}$ are C and B with the appropriate rows and columns removed. Then using Theorem 3.28 the number of vertex disjoint paths in G_M from excitations in $\mathcal{U}$ to nodes in $\mathcal{Y}$ is equal to the generic rank of $T_{\mathcal{Y}\mathcal{U}}(z,\theta)$. ■

4 Joint-direct identification of a network for any noise spectrum

4.1 Introduction

In this chapter based on (Weerts et al., 2018d) it is the objective to obtain a Maximum Likelihood estimate of all transfer functions in a dynamic network model, so the question addressed in this chapter is the following.

Can maximum likelihood estimates with minimum variance be obtained of a dynamic network for general noise conditions?

In single-input-single-output systems as (2.23) the prediction error method can be shown to be equal to Maximum Likelihood estimation under some conditions. When the identification setting is extended to multi-input-single-output identification of dynamic networks as in Section 2.3, then the relation to Maximum Likelihood has not been drawn in the literature. Adding flexible noise models that allow noise correlations and rank-reduced spectra make it more difficult to draw a parallel between prediction error methods and Maximum Likelihood estimates, since the prediction error methods in literature are not able to deal with these noises.

Correlated disturbances cause correlations between nodes that are not induced by the dynamic modules. This means that when correlated disturbances are not appropriately estimated, then estimated module dynamics are estimating the noise correlations as well as the original module dynamics. One way of dealing with correlated disturbances is formulated in (Dankers et al., 2017). The effect of correlated disturbances in a closed-loop system has been studied in (Van den Hof et al., 2017a). In that paper the objective is to estimate a single module consistently using a multi-input-single-output setting. Some noise correlations are blocked by including additional inputs, i.e. inputs that are not an in-neighbor to the output in the data generating network are modeled as an additional input. In this way some noise correlations can be explained by the additional module, preventing bias in the module of interest.

In this chapter an alternative approach to dealing with correlated noise is taken. Here we model additional outputs in the estimation problem, so the setting becomes multi-input-multi-output. The approach is named the *joint-direct method*, since the multiple outputs are predicted jointly. For this approach the prediction error reasoning of dynamic networks has to be revisited to jointly predict all the outputs. By modeling multiple outputs, the noise correlations can be modeled in a non-diagonal H model. A difficulty that appears with this approach is that rank-reduced noises are modeled with a non-square H.

Identification in the situation of rank-reduced noise is a topic that has not been widely addressed in the prediction error identification literature. Dynamic factor models have been developed in Deistler et al. (2015); Felsenstein (2014) to deal with rank-reduced noise. Maximum likelihood estimates with rank-reduced noise have been obtained for vector autoregressive systems (Kölbl, 2015) and linear regression (Srivastava and von Rosen, 2002). In classical multi-output prediction error methods (Ljung, 1999) rank-reduced noise has not been considered. A prediction error framework where no assumption on the rank of the noise is placed is to be developed, including the criterion, analysis of consistency, and the relation with Maximum Likelihood estimates. A preliminary study towards prediction error identification of a 1-input-2-output system with rank-reduced noise has been performed in (Van den Hof et al., 2017b).

Maximum Likelihood estimates are asymptotically efficient estimates where the variance asymptotically tends to the Cramér-Rao lower bound. Our approach is as follows: first we derive expressions for the variance of the estimated modules, in particular for the rank-reduced noise situation. With those expressions it can be analyzed under which conditions the variance is at the Cramér-Rao lower bound.

The chapter will proceed by defining the predictor, prediction error, and an identification criterion with consistency analysis for the situation of correlated noises in Section 4.2. In Section 4.3 another criterion specifically for the situation of rank-reduced noise is introduced, and maximum likelihood properties are analyzed. Then in Section 4.4 the variance is analyzed by investigating the Cramer-Rao lower bound. Finally, in Section 4.5, simulations show the benefit of appropriately taking rank-reduced noise into account.

4.2 Predictor, model set and WLS

In this section we develop a prediction error approach for the identification of dynamic networks. Before proceeding with the prediction error setup we must define which class of dynamic networks are being considered. In Chapter 3 conditions for identifiability have been presented, and we need to ensure that an identifiable model set can be constructed. The assumptions on the data generating network are formalized below.

> **Assumption 4.1.** *The data generating system $\mathcal{S}$ is represented by the network model M^0 where*
>
> - *the modules in $G^0(q)$ are strictly proper;*
> - *all nodes w are measured;*
> - *external excitation may be present;*
> - *the noise spectrum $\Phi_v(\omega)$ may be non-diagonal and is of rank $p \leq L$;*
> - *the nodes are ordered such that the first p nodes are affected by a rank p noise process.*

The requirement that all modules are strictly proper is applied also to the model set. In the model set a noise model is used where $H(q, \theta)$ and $\Lambda(q, \theta)$ are as defined in Chapter 2, and in particular they are not necessarily diagonal.

Remark 4.2. *We have considered the situation that all modules in $G(q, \theta)$ are strictly proper. This situation can be extended to the situation of having proper modules in $G(q, \theta)$, thus allowing direct feedthrough terms, as long as there are no algebraic loops in the network. In Chapter 5 we develop the prediction error approach for networks that have feedthrough or even algebraic loops. Since the formulation of the ML result will become technically more involved for non-strictly proper modules, we have preferred to restrict to the strictly proper module situation in the current chapter.*

The objective in this chapter is to estimate all modules in the network. Moreover, in order to model the correlations captured in the non-diagonal H, additional predictor outputs are to be included. For these reasons we define a multivariable predictor.

> **Definition 4.3.** *The one-step-ahead predictor for node signals $w(t)$ is defined as the conditional expectation*
>
> $$\hat{w}(t|t-1) := \mathbb{E}\left\{w(t) \mid w^{t-1},\ r^t\right\}, \tag{4.1}$$
>
> *conditioned on $w^{t-1} := \{w(0), w(1), \cdots, w(t-1)\}$ and $r^t := \{r(0), r(1), \cdots, r(t)\}$.*

This predictor definition is the classical predictor used in multivarable estimation (Söderström and Stoica, 1989; Ljung, 1999). In situations where noise is of a reduced rank, there are multiple ways to model the noise process v. In order to write a unique and explicit form for the predictor filters that generate the one-step-ahead prediction, we use the squared version of the noise model (2.9), i.e. $v = \check{H}^0\check{e}$ where $\check{H}$ is square and monic as discussed in Chapter 2. This leads to the following predictor expression.

Proposition 4.4. *For a dynamic network considered in* (2.4) *that has strictly proper modules, the one-step-ahead predictor of the node signals* $w(t)$ *is given by*

$$\hat{w}(t|t-1) = W_w^0(q)w(t) + W_r^0(q)r(t), \tag{4.2}$$

with the predictor filters

$$W_w^0(q) = I - (\breve{H}^0(q))^{-1}(I - G^0(q)), \tag{4.3}$$

$$W_r^0(q) = (\breve{H}^0(q))^{-1}R^0(q). \tag{4.4}$$

Proof. Provided in Appendix 4.7.1. ■

The classical predictor has been represented with a network model, instead of a MIMO open-loop transfer function. For situations where noise is full rank, the equality $\breve{H}^0 = H^0$ holds, so full rank noise will be treated as a special case throughout the chapter.

Remark 4.5. *In (Weerts et al., 2016c) the alternative noise model, determined by the non-square* $H^0(q)$ *was used as a basis for formulating the predictor filters. However due to intrinsic non-uniqueness of the corresponding filter expressions, the use of the square noise model* $\breve{H}^0$ *is more attractive. Note that a subtle difference between the noise models* $\breve{H}^0$ *and* H^0 *is that in* $\breve{H}^0$ *the feedthrough term of* H_b^0 *has been removed and is represented now in* $cov(\breve{e})$*.*

In the parameterized model the feedthrough of H_b is modeled by $\Gamma(\theta)$ defined as $\Gamma(\theta) := \lim_{z\to\infty} H_b(z, \theta)$. The transfer functions in predictor (4.2) are parameterized in accordance with the model set $\mathcal{M}$ to create the parameterized predictor

$$\hat{w}(t|t-1,\theta) = w(t) - \Big(\breve{H}(q,\theta)\Big)^{-1}\Big((I - G(q,\theta))w(t) - R(q,\theta)r(t)\Big), \tag{4.5}$$

with

$$\breve{H}(q,\theta) = \begin{bmatrix} H_a(q,\theta) & 0 \\ H_b(q,\theta) - \Gamma(\theta) & I \end{bmatrix}. \tag{4.6}$$

The prediction error is then defined as

$$\varepsilon(t,\theta) := w(t) - \hat{w}(t \mid t-1, \theta), \tag{4.7}$$

which is L-dimensional even in the rank-reduced case.

The typical identification criterion for multivariable predictor models is the Weighted Least Squares (WLS) criterion (Söderström and Stoica, 1989; Ljung, 1999)

$$\hat{\theta}_N^{WLS} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon^T(t,\theta)\; Q\; \varepsilon(t,\theta), \tag{4.8}$$

with $Q > 0$. Given the multivariate character of the prediction error, the WLS criterion will allow us to show maximimum likelihood properties, and thus asymptotic minimum variance properties of our estimated models in later sections. Because the prediction error has a different form than the prediction error in literature (Ljung, 1999; Söderström and Stoica, 1989), the consistency proofs from literature do not apply, and these results must be re-evaluated.

For analysis of the asymptotic properties of the parameter estimate (4.8) consider the asymptotic criterion

$$\theta^\star = \arg\min_{\theta\in\Theta} \bar{V}(\theta), \tag{4.9}$$

with

$$\bar{V}(\theta) = \bar{\mathbb{E}}\, \varepsilon^T(t,\theta)\, Q\, \varepsilon(t,\theta), \tag{4.10}$$

and $\bar{\mathbb{E}}$ defined as $\lim_{N\to\infty} \sum_{t=1}^{N} \mathbb{E}$, according to Ljung (1999). In classical literature it has been shown that the solution of the weighted least squares criterion converges to the solution of the asymptotic criterion under some mild conditions (Ljung, 1999). Based on this result we can formulate that, under the condition that $w(t)$ and $r(t)$ are jointly quasi-stationary, $r(t)$ is bounded, and $e(t)$ has bounded moments of order ≥ 4, it holds that

$$\hat{\theta}_N^{WLS} \to \theta^\star \text{ w.p. 1 as } N \to \infty. \tag{4.11}$$

With this convergence result, consistency is shown when $\theta^\star = \theta^0$. Conditions for consistency are formulated in the next proposition.

Proposition 4.6. *Consider data generated by a system that satisfies Assumption 4.1, and consider a model set $\mathcal{M}$ where all modules in G are strictly proper. Let $\theta^\star$ be defined by (4.9), then it holds that*[a]

$$\begin{aligned} &\{G(q,\theta^\star), H_a(q,\theta^\star), H_b(q,\theta^\star) - \Gamma(\theta^\star), R(q,\theta^\star)\} \\ &\quad = \{G^0(q), H_a^0(q), H_b^0(q) - \Gamma^0, R^0(q)\}, \end{aligned} \tag{4.12}$$

when the following conditions are satisfied

1. *The data generating system is in the model set, i.e. $\exists\theta_0 \in \Theta$ such that $M(\theta_0) = M^0$,*
2. *$\mathcal{M}$ is globally network identifiable at $M(\theta_0)$, and*
3. *the external excitation r, if present, is persistently exciting of sufficiently high order and uncorrelated to v.*

[a] Strictly speaking $\theta^\star$ can be a set and the equation holds for all $\theta \in \theta^\star$.

Proof. Provided in Appendix 4.7.2. ■

In this proposition the full rank noise situation appears as a special case. With full rank noise, (4.12) is written as

$$\{G(q,\theta^\star), H(q,\theta^\star), R(q,\theta^\star)\} = \{G^0(q), H^0(q), R^0(q)\}. \tag{4.13}$$

In the criterion (4.8) it can be observed that $H_b^0(q) - \Gamma^0$ is estimated consistently, but since Γ^0 is the feedthrough of $H_b^0(q)$, the Γ^0 is not estimated at all.

Information on Γ exists in the spectrum $\Phi_{\bar{v}}$ and in the residuals of the estimated model. For the situation of rank-reduced noise, when the modules G are strictly proper, then the feedthrough of $H_b(q,\theta)$ parameterized as $\Gamma(q,\theta)$ can directly be obtained from the noise spectrum $\Phi_{\bar{v}}(\omega)$ since

$$\Phi_{\bar{v}}(\infty) = (I - G^{\infty})^{-1} \begin{bmatrix} I \\ \Gamma \end{bmatrix} \Lambda \begin{bmatrix} I \\ \Gamma \end{bmatrix}^T (I - G^{\infty})^{-T}, \tag{4.14}$$

with $G^{\infty} = 0$. This implies that no structure has to be imposed onto $\Gamma(\theta)$. When the WLS criterion has been applied, then the Γ can also be estimated in the following way. Based on the dependencies in the innovation we split the prediction error into 2 parts:

$$\varepsilon(t,\theta) = \begin{bmatrix} \varepsilon_a(t,\theta) \\ \varepsilon_b(t,\theta) \end{bmatrix}, \tag{4.15}$$

where $\varepsilon_a \in \mathbb{R}^p$, and $\varepsilon_b \in \mathbb{R}^{L-p}$. Under zero initial conditions in the system and the predictor filters, the prediction error, when evaluated at $\theta = \theta_0$, has the same dependencies as the innovation, i.e.

$$\varepsilon_a(t,\theta_0) = e(t), \quad \text{and} \quad \varepsilon_b(t,\theta_0) = \Gamma^0 e(t),$$

such that $\Gamma^0 \varepsilon_a(t,\theta^0) = \varepsilon_b(t,\theta^0)$. Using this knowledge an estimation of Γ^0 can be made by

$$\hat{\Gamma}_N = \left(\frac{1}{N} \sum_{t=1}^{N} \varepsilon_b(\hat{\theta}_N) \varepsilon_a^T(\hat{\theta}_N) \right) \left(\frac{1}{N} \sum_{t=1}^{N} \varepsilon_a(\hat{\theta}_N) \varepsilon_a^T(\hat{\theta}_N) \right)^{-1}. \tag{4.16}$$

Since $\hat{\theta}_N$ is a consistent estimator, this estimate $\hat{\Gamma}_N$ will converge to

$$\Gamma^{\star} = \left(\mathbb{E}\, \varepsilon_b(\theta^{\star}) \varepsilon_a^T(\theta^{\star}) \right) \left(\mathbb{E}\, \varepsilon_a(\theta^{\star}) \varepsilon_a^T(\theta^{\star}) \right)^{-1} \tag{4.17}$$

which is

$$\Gamma^{\star} = \Gamma^0 \Lambda^0 (\Lambda^0)^{-1} = \Gamma^0. \tag{4.18}$$

For full-rank noise the weight $Q = (\Lambda^0)^{-1}$ typically leads to minimum variance estimates (Ljung, 1999), but for rank-reduced noise $\check{\Lambda}^0$ is not invertible. In order to obtain minimum variance properties, a new approach is needed to determine an appropriate weighting Q in the identification criterion (4.8).

The identification method that is presented in this section is termed as "joint-direct method", as it combines elements from two classical methods for closed-loop identification (Ljung, 1999), i.e. the joint-io method that is based on treating all measured signals jointly and starts with estimating closed-loop transfer function objects, and the direct method in which plant and noise dynamics are parametrized directly.

4.3 Constrained least squares and ML

4.3.1 Constrained least squares

When noise is rank-reduced, the WLS criterion does not take into account the fact that there are dependencies in the innovation process $\check{e}(t)$, as represented in (2.11). In this section an identification criterion is introduced which properly takes these dependencies into account. Maximum likelihood properties of the estimators are investigated, which implies an appropriate choice of weight for the WLS criterion. Based on the dependencies in the innovation we define

$$Z(t,\theta) := \Gamma(\theta)\varepsilon_a(t,\theta) - \varepsilon_b(t,\theta), \tag{4.19}$$

and introduce the Constrained Least Squares (CLS) criterion:

$$\begin{aligned} \hat{\theta}_N^{CLS} &= \arg\min_\theta \frac{1}{N}\sum_{t=1}^{N} \varepsilon_a(t,\theta)\ Q_a\ \varepsilon_a(t,\theta) \\ \text{subject to } &\frac{1}{N}\sum_{t=1}^{N} Z^T(t,\theta)Z(t,\theta) = 0, \end{aligned} \tag{4.20}$$

with $Q_a > 0$. For finite N, the quadratic constraint is equivalent to the constraint $Z(t,\theta) = 0\ \forall t$, which was introduced in Weerts et al. (2017). We have chosen for a quadratic constraint as this facilitates the convergence and consistency result in the next proposition, and because it is less computationally demanding.

While the term $\Gamma(\theta)$ was only estimated after optimizing the WLS criterion, in the CLS criterion it enters the estimation procedure directly through the constraint. Consistency of the CLS estimate can now be formulated in the next proposition of which a preliminary version was presented in Weerts et al. (2017).

Proposition 4.7. *Consider data generated by a system that satisfies Assumption 4.1, and consider a model set $\mathcal{M}$ where all modules in G are strictly proper. Let $\hat{\theta}_N^{CLS}$ be defined by (4.20) and let θ^* be defined by*

$$\begin{aligned} \theta^* &= \arg\min_\theta \bar{\mathbb{E}}\ \varepsilon_a(t,\theta)\ Q_a\ \varepsilon_a(t,\theta) \\ \textit{subject to } &\bar{\mathbb{E}}\ Z^T(t,\theta)Z(t,\theta) = 0. \end{aligned} \tag{4.21}$$

1. *Under the conditions that $w(t)$ and $r(t)$ are jointly quasi-stationary, $r(t)$ is bounded, and $e(t)$ has bounded moments of order ≥ 4, it holds that*

$$\hat{\theta}_N^{CLS} \to \theta^* \textit{ w.p. } 1 \textit{ as } N \to \infty. \tag{4.22}$$

2. *It holds that*[a]

$$\{G(q,\theta^*), H(q,\theta^*), R(q,\theta^*)\} = \{G^0(q), H^0(q), R^0(q)\}, \tag{4.23}$$

when the following conditions are satisfied

1. *The data generating system is in the model set, i.e. $\exists\theta_0 \in \Theta$ such that $M(\theta_0) = M^0$,*

2. *$\mathcal{M}$ is globally network identifiable at $M(\theta_0)$, and*

3. *the external excitation r, if present, is persistently exciting of sufficiently high order and uncorrelated to v.*

[a] Strictly speaking θ^* can be a set and the equation holds for all $\theta \in \theta^*$.

Proof. Provided in Appendix 4.7.3. ■

As opposed to the consistency result for the WLS estimate in Proposition 4.6, now the term $\Gamma(\theta)$, which is included in $H_b(q,\theta)$, is also estimated consistently directly in the criterion. Estimation of Γ is taken care of by the constraint, that constrains the parameter space in order to guarantee the (static) dependency among the terms of the prediction error.

4.3.2 Maximum Likelihood

Maximum Likelihood estimates minimize the variance of the estimated model. Motivated by the proper handling of the dependencies in the noise terms, it can be expected that the CLS estimate has a close resemblance with the Maximum Likelihood estimate. This is analysed next.

Theorem 4.8. *Let $e(t)$ be normally distributed and zero mean, i.e. $e(t) \sim \mathcal{N}(0, \Lambda^0)$, and consider a parameterized model set as in Definition 3.35. Then under zero initial conditions*[a]*:*

1. *The Maximum Likelihood estimate of θ^0 is*

$$\begin{aligned} \hat{\theta}_N^{ML} &= \arg\max_{\theta} \log L_a(\theta) \\ &\textit{subject to } \frac{1}{N}\sum_{t=1}^{N} Z^T(t,\theta)Z(t,\theta) = 0, \end{aligned} \tag{4.24}$$

with

$$\begin{aligned} \log L_a(\theta) = c &- \frac{N}{2}\log\det\Lambda(\theta) \\ &- \frac{1}{2}\sum_{t=1}^{N} \varepsilon_a^T(t,\theta)\Lambda^{-1}(\theta)\varepsilon_a(t,\theta). \end{aligned} \tag{4.25}$$

2. *Under the condition that $\Lambda(\theta)$ does not share parameters with $\varepsilon(t,\theta)$ the*

Maximum Likelihood estimate can alternatively be written as

$$\begin{aligned}\hat{\theta}_N^{ML} &= \arg\min_{\theta} \det\left(\frac{1}{N}\sum_{t=1}^{N}\varepsilon_a(t,\theta)\varepsilon_a^T(t,\theta)\right)\\ \textit{subject to } 0 &= \frac{1}{N}\sum_{t=1}^{N} Z^T(t,\theta)Z(t,\theta), \qquad (4.26)\\ \Lambda(\theta) &= \frac{1}{N}\sum_{t=1}^{N}\varepsilon_a(t,\theta)\varepsilon_a^T(t,\theta).\end{aligned}$$

[a] The zero initial conditions reflect values of input and output values of the predictor filters, prior to the time interval $[1, N]$, that are required to calculate the predicted node signal within the time interval.

Proof. Provided in Appendix 4.7.4. ■

In the theorem above two formulations of the Maximum Likelihood estimator are presented, and the second one deserves some attention. In (4.26) the last equation does not involve an actual constraint that limits the optimization problem, but it is merely there to specify the parameters that determine the estimated Λ. When the constraint on $\Lambda(\theta)$ is removed, then (4.26) may be used to estimate G, H, R without estimating Λ.

Note that when a model set with fixed (non-parameterized) Λ is used, then the term $-\frac{N}{2}\log\det\Lambda(\theta)$ in (4.24) becomes constant, and the Maximum Likelihood estimate (4.24) reduces to the Constrained Least Squares (4.20) estimate with $Q_a = \Lambda^{-1}$. This implies that the CLS equipped with the appropriate weight $Q_a = (\Lambda^0)^{-1}$ is a maximum likelihood estimator in case of Gaussian disturbances.

The maximum likelihood estimator also applies to situations of full rank noise as a special case. Then the estimator is

$$\begin{aligned}\hat{\theta}_N^{ML} &= \arg\min_{\theta} \det\left(\frac{1}{N}\sum_{t=1}^{N}\varepsilon(t,\theta)\varepsilon^T(t,\theta)\right)\\ \text{subject to } \Lambda(\theta) &= \frac{1}{N}\sum_{t=1}^{N}\varepsilon^T(t,\theta)\varepsilon(t,\theta).\end{aligned} \qquad (4.27)$$

For a fixed Λ this corresponds with the WLS (4.8) with $Q = \Lambda^{-1}$.

4.3.3 Practical situations

If initial conditions are non-zero and not explicitly dealt with in the parametrized model, then part of the prediction error is caused by the initial conditions. Although this effect asymptotically goes to 0, the ε_b does not have to be linearly dependent on ε_a, and consequently there do not exist parameters for which $Z(t,\theta) = 0$ for all t. Similarly in the situation where $\mathcal{M}$ does not contain $\mathcal{S}$, it is possible that there do not exist parameters for which $Z(t,\theta) = 0$ for all t. When the $Z(t,\theta)$ can not be made 0 the constraint in (4.20) and (4.24) is not feasible and the solution set of the criterion is empty.

In order to deal with situations where there are non-zero initial conditions, or where the system is not in the model set, we introduce a relaxed criterion. This relaxed criterion has a relaxed constraint, which appears as an additional penalty term, weighted by the real-valued penalty weight $\lambda > 0$:

$$\hat{\theta}_N^{rel} = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \Big(\varepsilon_a(t,\theta) Q_a \varepsilon_a(t,\theta) + \lambda Z^T(t,\theta) Z(t,\theta) \Big). \tag{4.28}$$

The above criterion is equivalent to the CLS (4.20) for $\lambda \to \infty$. Another way to write the relaxed criterion is as the WLS (4.8) with parameterized weight

$$Q(\theta) = \begin{bmatrix} Q_a + \lambda \Gamma^T(\theta)\Gamma(\theta) & -\lambda \Gamma^T(\theta) \\ -\lambda \Gamma(\theta) & \lambda I \end{bmatrix}. \tag{4.29}$$

In a situation where the number of samples N is finite, and where initial conditions are not estimated, the optimal prediction error may not be rank-reduced. Then the constraint of the CLS, which enforces a rank-reduced prediction error, causes bias. Relaxing the CLS by tuning parameter λ is effectively a trade-off between bias and variance. The optimal choice for λ will depend on the contribution of initial conditions, the contribution of unmodeled dynamics and the length of the data record. Determining the optimal bias-variance trade-off is not considered here.

4.4 Cramer-Rao lower bound

4.4.1 Variance of Weighted Least Squares Estimates - full rank

In the situation that the noise is full rank the classical parameter variance results (Ljung, 1999) can be applied. For $N \to \infty$ and $\mathcal{S} \in \mathcal{M}$ the estimate converges under weak conditions to a normal distribution given by

$$\underbrace{\sqrt{N}(\hat{\theta}_N^{CLS} - \theta^0)}_{:=\tilde{\theta}} \sim \mathcal{N}(0, P_\theta), \tag{4.30}$$

with P_θ positive definite. For full-rank noise processes, P_θ is defined by

$$\begin{aligned} P_\theta = \left[\bar{\mathbb{E}}\psi(t) Q \psi^T(t)\right]^{-1} \left[\bar{\mathbb{E}}\psi(t) Q \Lambda^0 Q \psi^T(t)\right] \cdot \\ \cdot \left[\bar{\mathbb{E}}\psi(t) Q \psi^T(t)\right]^{-1}, \end{aligned} \tag{4.31}$$

with

$$\psi(t) := -\frac{d}{d\theta}\varepsilon^T(t,\theta)|_{\theta=\theta_0}. \tag{4.32}$$

It can be shown that the weight $Q = (\Lambda^0)^{-1}$ leads to the minimum variance estimate Söderström and Stoica (1989).

The variance has some lower bound P_θ^0, $P_\theta \geq P_\theta^0$, which for full-rank noise is given by

$$P_\theta^0 = \left[\bar{\mathbb{E}}\psi(t)(\Lambda^0)^{-1}\psi^T(t)\right]^{-1}. \tag{4.33}$$

4.4.2 Variance of Weighted Least Squares Estimates - rank-reduced

For rank-reduced noise and the WLS criterion (4.8) the expression for P_θ is similar to the expression above, with Λ^0 replaced by $\breve{\Lambda}^0$. This can be shown by following its derivation in Söderström and Stoica (1989) and using $\breve{\Lambda}^0$ instead of Λ^0. For the rank-reduced case, Λ^0 would have to be replaced by $\breve{\Lambda}^0$, which is singular, and so its inverse does not exist. Therefore a lower bound like (4.33) is not valid in this case.

The question is now what the minimum variance is when noise is rank-reduced. In the following example we are looking for a weight Q which leads to minimum variance in a simple rank-reduced noise estimation problem.

Example 4.9. *Consider the system in Figure 4.1, where 2 parameters are to be estimated, θ_a and θ_b. The system is governed by*

$$\begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} = \begin{bmatrix} a^0 & 0 \\ 0 & b^0 \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} e(t) \\ e(t) \end{bmatrix}}_{\breve{e}(t)}.$$

The disturbance process $\breve{e}$ has covariance matrix $\breve{\Lambda}^0 = [\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}]$, which is singular. When

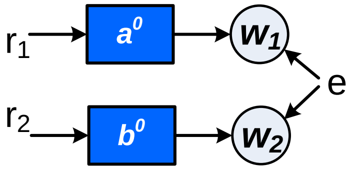


Figure 4.1: System with 2 nodes, no dynamics and 1 noise disturbance. It is excited by the quasi-stationary excitation signals r_1, r_2 and the stochastic process e which are all mutually uncorrelated and have unit variance.

the WLS (4.8) is used with a weight Q defined by (4.29) and $\Gamma(\theta)$ set to 1,

$$Q = \begin{bmatrix} 1+\lambda & -\lambda \\ -\lambda & \lambda \end{bmatrix} \tag{4.34}$$

with $\lambda > 0$ and prediction errors

$$\varepsilon_a = w_1 - \theta_a r_1, \quad \varepsilon_b = w_2 - \theta_b r_2, \tag{4.35}$$

then we get a consistent estimate. The identification criterion to be minimized becomes

$$\frac{1}{N}\sum_{t=1}^{N}\left\{\frac{1}{\lambda}\varepsilon_a^2(\theta_a) + \left(\begin{bmatrix} 1 & -1 \end{bmatrix}\begin{bmatrix} \varepsilon_a(\theta_a) \\ \varepsilon_b(\theta_b) \end{bmatrix}\right)^2\right\}. \tag{4.36}$$

In the limit as $\lambda \to \infty$, Q becomes singular, and the expression for the identification criterion becomes

$$\frac{1}{N}\sum_{t=1}^{N}\left((a^0 - \theta_a)r_1 + e - (b^0 - \theta_b)r_2 - e\right)^2, \tag{4.37}$$

which is obtained by substituting (4.35) *into* (4.36). *In this expression disturbance e drops out, and a variance-free estimate of* a^0 *and* b^0 *is obtained. This phenomenon of variance-free estimation has also been observed in (Everitt et al., 2015).*

This phenomenon should also be observed in the variance expression (4.31). To this end we use

$$\psi(t) = \begin{bmatrix} r_1(t) & 0 \\ 0 & r_2(t) \end{bmatrix}, \tag{4.38}$$

such that we obtain

$$\bar{\mathbb{E}}\psi Q \psi^T = \bar{\mathbb{E}} \begin{bmatrix} r_1^2(1+\lambda) & -r_1 r_2 \lambda \\ -r_1 r_2 \lambda & r_2^2 \lambda \end{bmatrix} = \begin{bmatrix} (1+\lambda) & 0 \\ 0 & \lambda \end{bmatrix} \tag{4.39}$$

and

$$\bar{\mathbb{E}}\psi Q \check{\Lambda}^0 Q \psi^T = \bar{\mathbb{E}} \begin{bmatrix} r_1^2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.40}$$

We can compute P_θ *of (4.31) as*

$$P_\theta = \begin{bmatrix} (1+\lambda) & 0 \\ 0 & \lambda \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (1+\lambda) & 0 \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{(1+\lambda)^2} & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.41}$$

Here we can see that as $\lambda \to \infty$ *the covariance goes to 0.*

It could be tempting to use an expression like (4.33) for the lower bound on the variance, with the inverse covariance $(\Lambda^0)^{-1}$ *replaced by a pseudo-inverse of* Λ^0. *In this example* $(\check{\Lambda}^0)^\dagger = \frac{1}{4}\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ *and substituting this into (4.33) instead of* $(\Lambda^0)^{-1}$, *delivers*

$$\left[\bar{\mathbb{E}}\psi(t)(\check{\Lambda}^0)^\dagger \psi^T(t)\right]^{-1} \neq 0, \tag{4.42}$$

which can not be the expression for the minimum variance.

Note that this example is fully symmetric in nodes w_1 and w_2, or equivalently in systems a^0 and b^0. Nevertheless one of the parameters θ_b is estimated variance-free, while θ_a is not. This is the result of the particular choice of weighting function, that according to (4.29) reflects the choice of w_1 as the full-rank noise node. Choosing the alternative weight $Q = \left[\begin{smallmatrix} \lambda & -\lambda \\ -\lambda & 1+\lambda \end{smallmatrix}\right]$ would resemble the situation of choosing w_2 as the full rank noise node. For both the weights, when we let $\lambda \to \infty$ the variance-free maximum likelihood estimate is obtained, which is again symmetric in θ_a and θ_b.

In this example it is possible to choose a weight beyond the structure of (4.29), e.g. $Q = \left[\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right]$, in which case we arrive at a variance-free estimate, since $Q\Lambda^0 Q = 0$. For this choice of Q we are essentially only modeling the 'constraint', and we dropped the 'original cost function' ε_a^2. Such a weight Q is useful when all parameters in the model can be estimated using just the constraint.

The conclusion that can be drawn from the example is we have two options to minimize variance in a rank-reduced estimation problem. Either we use the CLS criterion, or we must use the WLS with weight described in (4.29). Expressions for the variance of the WLS and CLS criteria are derived in the next section.

4.4.3 Variance of Constrained Least Squares Estimates

In this section we will derive a closed-form expression for the variance of the CLS estimates. We must address the full impact of the constraint, that typically reduces the effective parameter space in the criterion.

For the CLS situation the asymptotic identification criterion is written as follows:

$$\begin{aligned}\theta^* = &\arg\min_\theta \bar{\mathbb{E}}\ \varepsilon_a(t,\theta)\ Q_a\ \varepsilon_a(t,\theta)\\ &\text{subject to: } \bar{\mathbb{E}}\ Z^T(t,\theta)Z(t,\theta) = 0.\end{aligned} \tag{4.43}$$

We are making an analysis of the variance around the data generating system parameters, so we assume that $\theta^* = \theta^0$.

In a neighborhood around $\theta = \theta^0$ the constraint can be approximated using a first order Taylor series

$$Z(t,\theta) \approx Z(t,\theta^0) + \frac{\partial Z(t,\theta)}{\partial\theta}\Big|_{\theta=\theta^0}(\theta-\theta^0) = A(t)(\theta-\theta^0), \tag{4.44}$$

where

$$A(t) := \frac{\partial Z(t,\theta)}{\partial\theta}\Big|_{\theta=\theta^0} \tag{4.45}$$

with $A \in \mathbb{R}^{(L-p)\times n_\theta}$. The approximated constraint is then

$$\bar{\mathbb{E}}\ (\theta-\theta^0)^T A^T(t)A(t)(\theta-\theta^0) = 0, \tag{4.46}$$

where $\bar{\mathbb{E}}\ A^T(t)A(t)$ is of dimension $n_\theta \times n_\theta$. Note that $Z(t,\theta^0) = 0$, but that $Z(t,\theta)$ with θ in the neighborhood around θ^0 is non-zero. Similarly the $A(t)(\theta-\theta^0)$ is 0 for $\theta = \theta^0$ and non-zero in the neighborhood, which implies that the parameter space that is being constrained by (4.46) is the same as in the CLS criterion. We can define a matrix Π of dimension $(n_\theta - n_\rho) \times n_\theta$ such that the expectation is

$$\bar{\mathbb{E}}\ A^T(t)A(t) = \Pi^T\Pi, \tag{4.47}$$

where matrix Π has no particular structure. Then in the neighborhood of the estimate θ^* the constraint is approximated by a quadratic constraint

$$\begin{aligned}\theta^* = &\arg\min_\theta \bar{\mathbb{E}}\ \varepsilon_a(t,\theta)\ Q_a\ \varepsilon_a(t,\theta)\\ &\text{subject to: } (\theta-\theta^0)^T\Pi^T\Pi(\theta-\theta^0) = 0.\end{aligned} \tag{4.48}$$

In order to appropriately take the constraint into account in the variance analysis, a re-parameterization will be considered using a parameter ρ with $\dim(\rho) = n_\rho < \dim(\theta)$. The two parameters will be related through a mapping induced by the constraint, such that the new parameterization trivially satisfies the constraint.

Lemma 4.10. *The constrained parameter space determined by* $(\theta-\theta^0)^T\Pi^T\Pi(\theta-\theta^0)=0$*, with* Π *defined as above, is equivalently described by*

$$\theta = S\rho + C, \quad \textit{with } \rho \in \mathbb{R}^{n_\rho}, \tag{4.49}$$

where $S \in \mathbb{R}^{n_\theta \times n_\rho}$ *satisfies* $\Pi S = 0$ *and is full column rank, i.e.* S *characterizes the right nullspace of* Π*, and* $C = \Pi^\dagger\Pi\theta^0$*, where* $\Pi^\dagger$ *satisfies* $\Pi\Pi^\dagger = I$*.*

Proof. Provided in Appendix 4.7.5. ■

The unconstrained parameter ρ can now be used to rewrite the criterion (4.48) into a form that trivially satisfies the constraint. The resulting criterion is then essentially an unconstrained criterion operating on a lower dimensional parameter ρ.

Proposition 4.11. *The optimization problem (4.48) can equivalently be written as*

$$\theta^* = S\rho^* + C, \tag{4.50}$$

with

$$\rho^* = \arg\min_\rho \bar{\mathbb{E}}\, \varepsilon_a(t, S\rho + C)\ Q_a\ \varepsilon_a(t, S\rho+C). \tag{4.51}$$

Proof. Provided in Appendix 4.7.6. ■

Since (4.51) is an unconstrained identification criterion, we know that the asymptotic variance of the estimate $\hat{\rho}_N$ that corresponds to the asymptotic estimate ρ^* is given by

$$\begin{aligned} P_\rho = \left[\bar{\mathbb{E}}\psi_\rho(t)Q_a\psi_\rho^T(t)\right]^{-1} &\left[\bar{\mathbb{E}}\psi_\rho(t)Q_a\Lambda^0Q_a\psi_\rho^T(t)\right]\cdot \\ &\cdot\left[\bar{\mathbb{E}}\psi_\rho(t)Q_a\psi_\rho^T(t)\right]^{-1}, \end{aligned} \tag{4.52}$$

with

$$\psi_\rho(t) = -\frac{d}{d\rho}\varepsilon_a^T(t, S\rho + C)|_{\rho=\rho^*}. \tag{4.53}$$

Combining this expression with (4.49) now provides an expression for P_θ, as formulated next.

Proposition 4.12. *The covariance matrices* P_ρ *and* P_θ *satisfy the following relation*

$$P_\theta = SP_\rho S^T. \tag{4.54}$$

Proof. Provided in Appendix 4.7.7. ■

It is well known that the lower bound of P_ρ is achieved when $Q_a = (\Lambda^0)^{-1}$, such that

$$P_\rho \geq P_\rho^0 = \left[\bar{\mathbb{E}}\psi_\rho(t)(\Lambda^0)^{-1}\psi_\rho^T(t)\right]^{-1}. \tag{4.55}$$

Then by Proposition 4.12 the lower bound of P_θ is

$$P_\theta \geq P_\theta^0 = S P_\rho^0 S^T, \tag{4.56}$$

which is achieved for $Q_a = (\Lambda^0)^{-1}$.

Matrix S characterizes the right-nullspace of Π, so it is not a unique matrix. The covariance matrix P_ρ is a function of S and C, which makes it that all possible S matrices lead to the same P_θ and lower bound P_θ^0. As an illustration of the results, an Example is shown for the CLS estimate.

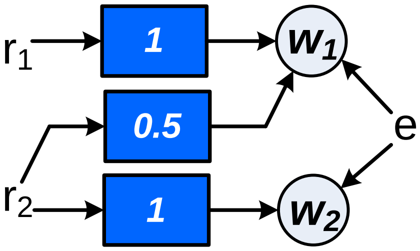


Figure 4.2: System with 2 nodes, no dynamics and 1 noise disturbance. It is excited by the quasi-stationary excitation signals r_1, r_2 and the stochastic process e which are all mutually uncorrelated and have unit variance.

Example 4.13. *In this example, depicted in Figure 4.2, the system is given by*

$$w_1(t) = r_1(t) + 0.5 r_2(t) + e(t), \quad w_2(t) = r_2(t) + e(t). \tag{4.57}$$

The noise is rank reduced, and has covariance matrix $\check{\Lambda}^0 = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, *which is singular. When the CLS (4.20) is used with knowledge of* $\Gamma^0 = 1$ *and prediction errors*

$$\varepsilon_1 = w_1 - \theta_{a1} r_1 - \theta_{a2} r_2, \quad \varepsilon_2 = w_2 - \theta_b r_2, \tag{4.58}$$

where $\varepsilon_1 = \varepsilon_a$ *and* $\varepsilon_2 = \varepsilon_b$, *then we get a consistent estimate. The constraint here consists of* $\bar{\mathbb{E}}\, Z^2(t,\theta) = 0$ *with*

$$Z(t,\theta) = \varepsilon_1(t,\theta) - \varepsilon_2(t,\theta). \tag{4.59}$$

Determining the approximated constraint requires taking the derivative

$$A(t) = \frac{\partial Z(t,\theta)}{\partial \theta}\bigg|_{\theta=\theta^*} = \begin{bmatrix} -r_1(t) & -r_2(t) & r_2(t) \end{bmatrix}. \tag{4.60}$$

When evaluating the expectation in the constraint we have

$$\bar{\mathbb{E}} A^T(t) A(t) = \bar{\mathbb{E}} \begin{bmatrix} r_1^2(t) & 0 & 0 \\ 0 & r_2^2(t) & -r_2^2(t) \\ 0 & -r_2^2(t) & r_2^2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \tag{4.61}$$

which can be factorized into $\Pi^T \Pi$ *with* $\Pi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix}$.

Vectors S and C can now be determined based on $\Pi S = 0$ and $C = \Pi^\dagger \Pi \theta^0$, leading to:

$$S = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ -0.5 \\ 0 \end{bmatrix}.$$

With this choice of S, we can determine ψ_ρ using (4.53) as

$$\psi_\rho = -\frac{d}{d\rho}\big(w_1 - \begin{bmatrix} r_1 & r_2 & 0 \end{bmatrix}(S\rho + C)\big) = r_2 \tag{4.62}$$

Then P_ρ of (4.52) is given by:

$$P_\rho = (\bar{\mathbb{E}}\, r_2^2)^{-1}(\bar{\mathbb{E}}\, r_2^2)(\bar{\mathbb{E}}\, r_2^2)^{-1} = 1,$$

where $\Lambda^0 = Q_a = 1$. Then with Proposition 4.12 the covariance of θ is determined as

$$P_\theta = SP_\rho S^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Since we used the optimal weighting $Q_a = (\Lambda^0)^{-1}$ this is also the lower bound on the variance in the given situation. Note that in the considered situation the first parameter θ_{a_1} is estimated variance-free.

It becomes interesting to analyze when it is beneficial to model a rank-reduced noise. The most clear case is in a situation where the matrix $\bar{\mathbb{E}}A^T(t)A(t)$ is square and full rank, as then the constraint uniquely determines all parameters, and *all* parameters are determined variance-free. In other situations some individual modules may be estimated variance-free, or the variance can be reduced. These results suggest that in a situation where noise is dominated by a few sources, and the other noise sources are small, then it may be beneficial to model this as rank-reduced noise in order to reduce variance.

4.4.4 Comparison to other work

Using a different reasoning than presented above, and not related to dynamic networks, in Stoica and Ng (1998) the Cramér-Rao lower bound on the variance under parametric constraints has been derived for Gaussian distributed noise. That result can be linked to the lower bound obtained in the previous section. In Stoica and Ng (1998) it is stated that first the Fisher information matrix J of the unconstrained part of the criterion (4.20) is obtained, which is

$$J = \bar{\mathbb{E}}\, \psi_a(t)\Lambda_0^{-1}\psi_a^T(t), \tag{4.63}$$

with $\psi_a(t) = \psi(t)\left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right]$. This unconstrained part of the criterion does not contain all parameters, meaning that ψ_a contains rows that are 0, and J is singular. The lower bound on the variance can not be given by J^{-1} since it does not exist. In Stoica and Ng (1998) it has been proven that the lower bound is given by

$$P_\theta^0 = S\left(S^T\bar{\mathbb{E}}\,\psi_a(t)\Lambda_0^{-1}\psi_a^T(t)S\right)^{-1}S^T, \tag{4.64}$$

with S as defined before. The above expression is equal to the lower bound in (4.56) that we obtained using a different reasoning, since by the chain rule for differentiation we have that

$$\psi_\rho(t) = S^T\psi_a(t) \tag{4.65}$$

which can be substituted in (4.64) to arrive at (4.56).

4.5 Simulations

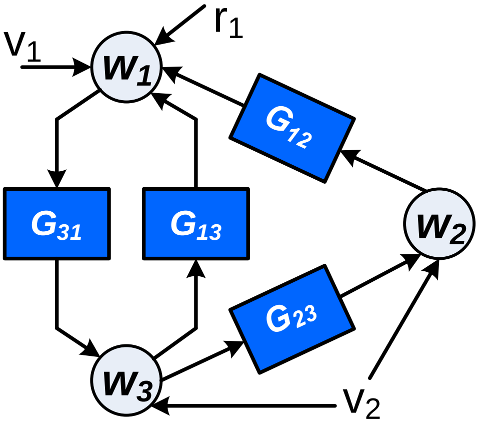


Figure 4.3: Example of a network with rank-reduced noise. Node signals are w_i, being the outputs of the (circular) summation points, interconnected by modules G_{ij} and perturbed by non-measured disturbance signals v_i. Signals r_i are excitation signals available to the user.

In this simulation example a 3 node network will be identified from data using the WLS and CLS criteria. We use the network in Figure 4.3 with $r_2 = 0$ and v a 2-dimensional white noise process with $\Lambda^0 = I$, such that

$$G^0 = \begin{bmatrix} 0 & G_{12}^0 & G_{13}^0 \\ 0 & 0 & G_{23}^0 \\ G_{31}^0 & 0 & 0 \end{bmatrix}, \quad H^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

The dynamic modules are finite impulse responses with the following coefficients

$$\begin{bmatrix} G_{12}^0(q) \\ G_{13}^0(q) \\ G_{23}^0(q) \\ G_{31}^0(q) \end{bmatrix} = \begin{bmatrix} 0.33 & -0.2 & 0.13 & -0.08 & 0.05 \\ 0.2 & -0.45 & -0.73 & -0.54 & -0.25 \\ -0.15 & 0.12 & -0.9 & 0.6 & 0.3 \\ -0.5 & 0.06 & -0.1 & 0.03 & 0 \end{bmatrix} \begin{bmatrix} q^{-1} \\ q^{-2} \\ q^{-3} \\ q^{-4} \\ q^{-5} \end{bmatrix}.$$

In total 100 Monte-Carlo simulations are performed on the above network with $N = 1000$ samples taken for each data set, and with initial conditions set to 0.

A model structure is used with $G(q,\theta)$ having the same structure as G^0, $H(q,\theta) = \left[\begin{smallmatrix} I \\ \Gamma(\theta_\Gamma) \end{smallmatrix}\right]$, and with $\Lambda = I$. Parameters are collected in the vector

$$\theta^T = \begin{bmatrix}\theta_{12}^T & \theta_{13}^T & \theta_{23}^T & \theta_{31}^T & \theta_\Gamma^T\end{bmatrix} \in \mathbb{R}^{22}, \tag{4.66}$$

where θ_{ij} correspond to module $G_{ij}(\theta_{ij})$. The prediction error can be denoted by

$$\begin{bmatrix}\varepsilon_1(t,\theta)\\ \varepsilon_2(t,\theta)\\ \varepsilon_3(t,\theta)\end{bmatrix} = \begin{bmatrix}w_1(t)\\ w_2(t)\\ w_3(t)\end{bmatrix} - \begin{bmatrix}0 & \phi_2(t) & \phi_3(t) & 0\\ 0 & 0 & \phi_3(t) & 0\\ \phi_1(t) & 0 & 0 & 0\end{bmatrix}\theta, \tag{4.67}$$

with appropriately chosen regressors $\phi_i(t)$.

The WLS is applied as the relaxed CLS with weight (4.29) parameterized with $\Gamma(\theta)$. The prediction errors are linear in the parameters, so the WLS criterion is straightforward to implement as a linear regression problem. In order to solve the CLS criterion a constraint optimization has been implemented and solved using Matlab's `fmincon()` function. Two different choices for λ are used to illustrate the effect of increasing values of λ. Results of the WLS estimates, and of the CLS estimates, are plotted in Figure 4.4.

It can be observed that the parameters of modules G_{12} and G_{13} do not change with different criteria. The noise on node 1 is independent of noise on nodes 2 and 3, such that estimation of the node 1 parameters is essentially not affected by the constraint. The parameters of G_{23} and G_{31} are estimated with smaller variance when λ increases, since the estimate gets closer to the ML estimate. The parameters of Γ (indexed by numbers 21 and 22) are estimated with very small variance, even for small λ.

For this estimation the lower bound on the variance can be computed. An approximation of the constraint is made by taking the derivative of the constraint with respect to the parameters. The constraint is formulated as

$$Z(\theta) := \Gamma_1(\theta)\varepsilon_1(t,\theta) + \Gamma_2(\theta)\varepsilon_2(t,\theta) - \varepsilon_3(t,\theta). \tag{4.68}$$

Its derivative with respect to the parameters is

$$\begin{aligned} A(t) = -\frac{d}{d\theta}Z(t,\theta)\Big|_{\theta=\theta^0} &= \\ \begin{bmatrix}\Gamma_1^0\phi_2 & \Gamma_1^0\phi_3 & \Gamma_2^0\phi_3 & -\phi_1 & \phi_2\theta_{12}^0 + \phi_3\theta_{13}^0 & \phi_3\theta_{23}^0\end{bmatrix}, \end{aligned} \tag{4.69}$$

where $\Gamma^0 = \begin{bmatrix}\Gamma_1^0 & \Gamma_2^0\end{bmatrix}$. Then to build the matrix Π of maximum rank we need 12 time samples

$$\Pi = \begin{bmatrix}A(1)\\ \vdots\\ A(12)\end{bmatrix}. \tag{4.70}$$

Because of the fact that $\Gamma_1^0 = 0$, the Π matrix is structured such that the left most 10 columns are 0. The other 12 columns constitute a 12×12 matrix of full rank. Matrix

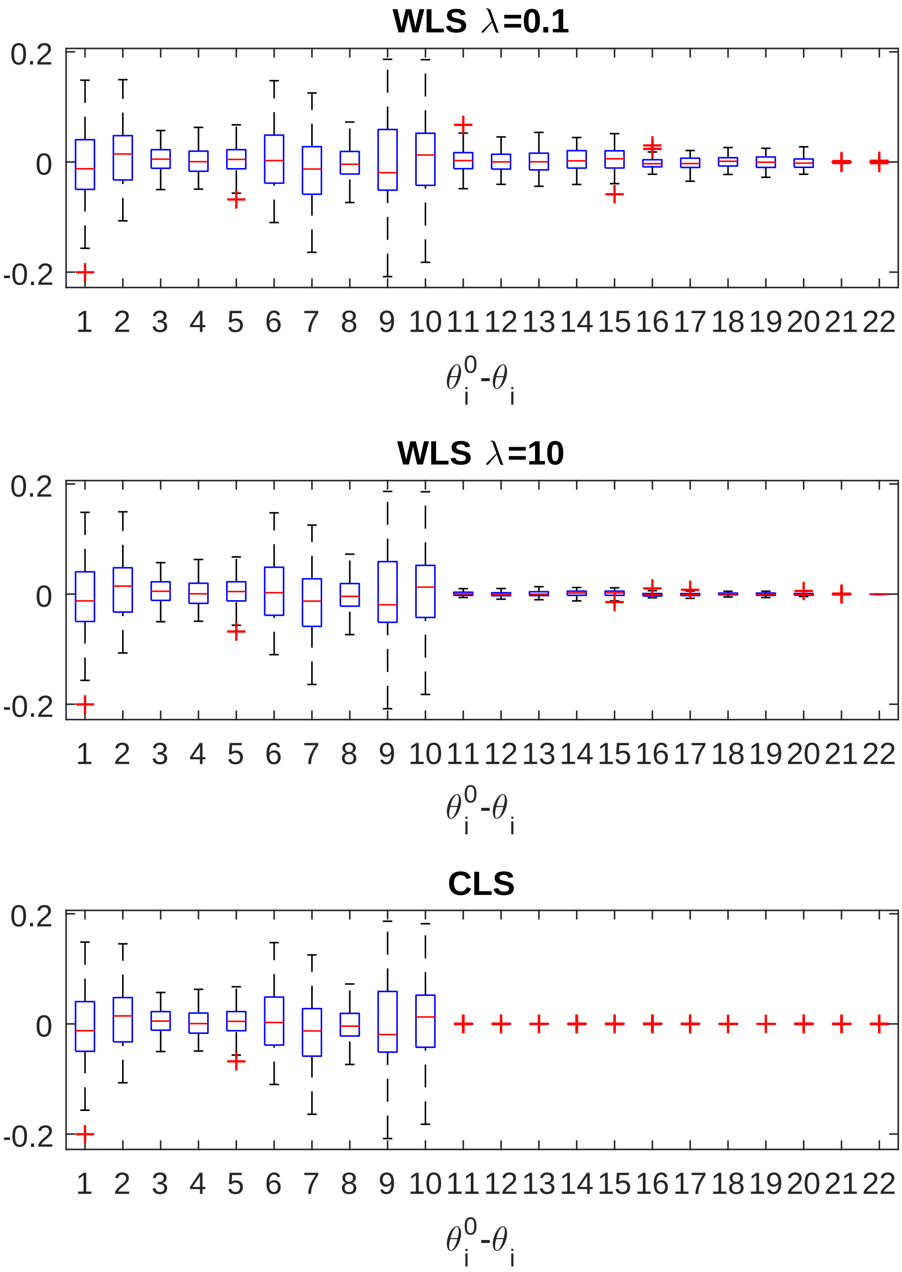


Figure 4.4: Boxplot of parameter estimation errors for the 22 different parameters over 100 Monte-Carlo runs. The top and middle figures are the WLS estimates (4.8) with weight (4.29) and $\lambda = 0.1$ and $\lambda = 10$ respectively, the bottom figure is the CLS estimate (4.20).

S is then defined by the right-nullspace of Π, and S has the particular structure that the first 10 rows are non-zero and form a 10×10 matrix of full rank, and the other 12 rows are 0 such that $\Pi S = 0$. When we consider $P_\theta^0 = SP_\rho^0 S^T$ and the structure of S, it is immediately observed that the lower bound on the variance of parameters 11 to 22 is 0.

The example above shows a similar phenomenon as the static Example 4.9, i.e. two modules that map into node variables that are subject to the same disturbance, and as a result of this are estimated variance-free.

4.6 Conclusions

Dynamic network models with rank-reduced noise can consistently be estimated under standard conditions when correlated and rank-reduced noise is appropriately included in the model set. In order to appropriately take the rank-reduced noise into account in the identification criterion, a weighted quadratic criterion subject to a constraint is utilized. Under fairly standard conditions this constrained criterion can be shown to result in maximum likelihood estimates. A classical variance expression can be derived for the weighted least squares estimator, but for the criterion with constraint the variance expressions is modified to appropriately take the constraint into account. For this latter situation explicit expressions for the variance have been derived, as well as expressions for the lower bound of this variance, reaching the Cramér- Rao lower bound for normally distributed noise.

Maximum Likelihood estimates are significant since these obtain the minimum amount of variance that is possible for an unbiased estimate. This implies that other asymptotically unbiased estimates of all network dynamics will not be able to improve over the joint-direct method in terms of variance. Moreover the joint-direct method is a suitable candidate method to be extended for tackling other network identification objectives. The work in this chapter may be extended to the identification of a single module, or for identification of the network topology.

4.7 Appendix

4.7.1 Proof of Proposition 4.4

First one predictor expression is derived using the square and monic noise model $\check{H}^0$, then it is shown that this is unique. We write the network equation (2.4) as

$$w = G^0 w + R^0 r + (\check{H}^0 - I)\check{e} + \check{e}.$$

Then we substitute using $He = \check{H}\check{e}$ and (2.4) the expression

$$\check{e} = (\check{H}^0)^{-1}[(I - G^0)w - R^0 r]$$

into the expression $(\check{H}^0 - I)\check{e}$, leading to

$$w = [I - (\check{H}^0)^{-1}(I - G^0)]w + (\check{H}^0)^{-1}R^0 r + \check{e}. \tag{4.71}$$

Since we assume that G^0 is strictly proper, $[I - (\check{H}^0)^{-1}(I - G^0)]$ is strictly proper, and evaluating the conditional expectation (4.1) leads to (4.2).

Now it is shown that the predictor filters W_w^0 and W_r^0 are unique. The predictor can be written as

$$\hat{w}(t|t-1) = w - \check{e} = W_w^0 w + W_r^0 r. \tag{4.72}$$

By subtracting $W_w^0 w$ from both sides, and substituting (2.1) we obtain

$$(I - W_w^0)(I - G^0)^{-1}(H^0 e + R^0 r) = \check{e} + W_r^0 r. \tag{4.73}$$

Now since e is independent from r we have

$$(I - W_w)(I - G)^{-1}He = \check{e}, \tag{4.74}$$

and then

$$(I - W_w)(I - G)^{-1} = \check{H}^{-1} \tag{4.75}$$

must hold. Since $(I - G)^{-1}$ and $\check{H}^{-1}$ are full rank, the expression for W_w is unique. From (4.73) we obtain $(I - W_w)(I - G)^{-1}Rr = W_r r$ since e is independent from r, where substituting (4.3) leads to the unique definition (4.4). ■

4.7.2 Proof of Proposition 4.6

First it will be shown that θ_0 is a minimum of the criterion, i.e. $\theta^0 \in \theta^\star$, after which it will be shown that $M(\theta_0)$ is the only minimum, i.e. $M(\theta_0) = M(\theta)\ \forall\ \theta \in \theta^\star$.

When combining (4.7), (4.5) and (2.4) it can be shown that the prediction error can be rewritten in terms of e and r

$$\begin{bmatrix} \varepsilon_a(\theta) \\ \varepsilon_b(\theta) \end{bmatrix} = F_e(q,\theta)e + \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} e + F_r(q,\theta)r, \tag{4.76}$$

with

$$F_e(\theta) := \check{H}^{-1}(\theta)(I - G(\theta))(I - G^0)^{-1}H^0 - \left[\begin{smallmatrix} I \\ \Gamma^0 \end{smallmatrix}\right],$$
$$F_r(\theta) := \check{H}^{-1}(\theta)\Big((I - G(\theta))(I - G^0)^{-1}R^0 - R(\theta)\Big),$$

where F_e is strictly proper since the innovation $\left[\begin{smallmatrix} I \\ \Gamma^0 \end{smallmatrix}\right] e$ has been written as a separate term.

The first term has a strictly proper filter, the innovation (second) term does not have delay, and since e is a white noise, the first 2 terms are uncorrelated with each other. By condition *2* the r term is uncorrelated with the e terms. In the quadratic function $\bar{V}(\theta)$ defined by (4.10) any cross-term between the 3 terms is 0 due to uncorrelatedness, therefore each of the terms can be minimized individually. Due to condition *1* the first and third terms are minimized by θ_0 and become 0. The second term does not contain parameters, so it is trivially minimized. Then we can conclude that $\theta^0 \in \theta^\star$.

Now it will be shown that any parameter θ_1 which reaches the minimum of the cost function must result in $M(\theta_0) = M(\theta_1)$. It can be shown ((Ljung, 1999) proof of Theorem 8.3) that

$$\begin{aligned} 0 &= \bar{V}(\theta_0) - \bar{V}(\theta_1) \\ &= \bar{\mathbb{E}}(\varepsilon(t,\theta_0) - \varepsilon(t,\theta_1))^T Q(\varepsilon(t,\theta_0) - \varepsilon(t,\theta_1)). \end{aligned} \tag{4.77}$$

Since $Q > 0$ we must have $\varepsilon(t,\theta_0) = \varepsilon(t,\theta_1)$, up to a possible transient term due to initial conditions, which decays to zero and therefore can be neglected in our asymptotic criterion. By condition *2*, $\begin{bmatrix} e(t) \\ r(t) \end{bmatrix}$ is a full rank process, such that

$$F_e(q,\theta_0) + \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} = F_e(q,\theta_1) + \begin{bmatrix} I \\ \Gamma^0 \end{bmatrix} \tag{4.78}$$

and

$$F_r(q,\theta_0) = F_r(q,\theta_1). \tag{4.79}$$

Since $F_e(q,\theta_0) = 0$ and $F_r(q,\theta_0) = 0$ we can write

$$\begin{bmatrix} I & 0 \\ \Gamma^0 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \Gamma^0 & 0 \end{bmatrix} + \begin{bmatrix} F_e(q,\theta_1) & F_r(q,\theta_1) \end{bmatrix}. \tag{4.80}$$

When we use the expressions for F_e and F_r, then pre-multiply both sides of (4.80) with $(I - G(q,\theta_1))^{-1}\check{H}(q,\theta_1)$, and finally add $\begin{bmatrix} 0 & (I - G(q,\theta_1))^{-1}R(q,\theta_1) \end{bmatrix}$ to both sides, then

$$\begin{aligned} &(I - G(\theta_0))^{-1} \begin{bmatrix} H(\theta_0) & R(\theta_0) \end{bmatrix} = \cdot \\ &\underbrace{\cdot\, (I - G(\theta_1))^{-1} \begin{bmatrix} H_a(\theta_1) & R_a(\theta_1) \\ H_b(\theta_1) - \Gamma(\theta_1) + \Gamma^0 & R_b(\theta_1) \end{bmatrix}}_{:=T'(\theta_1)} \end{aligned} \tag{4.81}$$

is obtained, where R_a and R_b are defined by $R(q,\theta) = \begin{bmatrix} R_a(q,\theta) \\ R_b(q,\theta) \end{bmatrix}$. Note that $\Gamma(\theta_1)$ is the feedthrough of $H_b(\theta_1)$, such that the feedthrough of H_b is being 'replaced' with the true values Γ^0, and $\Gamma(\theta_1)$ does not appear in the equation.

In (4.12) we make no claims on the feedtrough of H_b, we have to show that

$$\begin{aligned} &T'(\theta_1) = T'(\theta_0) \Rightarrow \\ &\{G(q,\theta^\star), H_a(q,\theta^\star), H_b(q,\theta^\star) - \Gamma(\theta^\star), R(q,\theta^\star)\} \\ &\qquad = \{G^0(q), H_a^0(q), H_b^0(q) - \Gamma^0, R^0(q).\} \end{aligned} \tag{4.82}$$

If we consider $\Theta' \in \Theta$ defined by all θ for which $\Gamma(\theta) = \Gamma^0$, then using the model set

$$\mathcal{M}' := \{M(\theta),\ \theta \in \Theta'\} \subseteq \mathcal{M},$$

we have that $T'(\theta) = T(\theta)$ for all $\theta \in \Theta'$. This means that we can apply the network identifiability reasoning to this situation. Since $\mathcal{M}'$ is a subset of $\mathcal{M}$, $\mathcal{M}'$ is globally network identifiable at $M(\theta_0)$ if $\mathcal{M}$ is is globally network identifiable at $M(\theta_0)$. Using condition *3* we then have that

$$\begin{array}{l} T'(q,\theta_1) = T'(q,\theta_0) \\ \qquad\qquad \Downarrow \\ \qquad\qquad \left\{ \begin{array}{ll} G(q,\theta_1) & = G^0(q) \\ H_a(q,\theta_1) & = H_a^0(q) \\ H_b(q,\theta_1) - \Gamma(\theta_1) & = H_b^0(q) - \Gamma^0 \\ R(q,\theta_1) & = R^0(q). \end{array} \right\} \end{array} \tag{4.83}$$

■

4.7.3 Proof of Proposition 4.7

The convergence proof in Ljung (1999) needs to be adapted slightly in order to prove (4.22). Under the conditions in part *(1)* the cost function converges

$$\sup_{\theta\in\Theta} \left| \frac{1}{N} \sum_{t=1}^{N} \varepsilon_a(t,\theta) Q_a \varepsilon_a(t,\theta) - \bar{\mathbb{E}}\varepsilon_a(t,\theta) Q_a \varepsilon_a(t,\theta) \right| \to 0 \tag{4.84}$$

*w.p.*1 as $N \to \infty$. Similarly the constraint converges

$$\sup_{\theta\in\Theta} \left| \frac{1}{N} \sum_{t=1}^{N} Z^T(t,\theta) Z(t,\theta) - \bar{\mathbb{E}} Z^T(t,\theta) Z(t,\theta) \right| \to 0 \tag{4.85}$$

w.p. 1 as $N \to \infty$. Since the cost and constraint in (4.20) both converge (4.22) must hold.

Using the same reasoning as the proof of Proposition 4.6, θ_0 is a minimum of the cost function, and θ_0 satisfies the constraint. Now it is shown that $M(q,\theta_0)$ is the only model that is a minimum of the cost function that satisfies the constraint, i.e. $M(q,\theta_0) = M(\theta)\ \forall\ \theta \in \theta^*$.

It can be shown ((Ljung, 1999) proof of Theorem 8.3) that

$$0 = \bar{\mathbb{E}}\varepsilon_a(t,\theta_0)^T Q_a \varepsilon_a(t,\theta_0) - \bar{\mathbb{E}}\varepsilon_a(t,\theta_1)^T Q_a \varepsilon_a(t,\theta_1) \tag{4.86}$$

if and only if

$$0 = \bar{\mathbb{E}}(\varepsilon_a(t,\theta_0) - \varepsilon_a(t,\theta_1))^T Q_a (\varepsilon_a(t,\theta_0) - \varepsilon_a(t,\theta_1)). \tag{4.87}$$

For the constraint we can use the fact that

$$Z(t,\theta_0) = \Gamma(\theta_0)\varepsilon_a(t,\theta_0) - \varepsilon_b(t,\theta_0) = 0, \quad \forall t \tag{4.88}$$

up to a possible transient term due to initial conditions that can be neglected in our asymptotic analysis. We can then rewrite the asymptotic constraint

$$0 = \bar{\mathbb{E}} Z^T(\theta_1) Z(\theta_1) \tag{4.89}$$

into the same form as (4.87)

$$0 = \bar{\mathbb{E}}(Z(\theta_0) - Z(\theta_1))^T (Z(\theta_0) - Z(\theta_1)). \tag{4.90}$$

Due to condition *(b)* and $Q_a > 0$ the predictor filters are identified from the above two equations, using the definitions of F_e and F_r from the proof of Proposition 4.6

$$\left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_0) & -I \end{smallmatrix}\right] \left(F_e(\theta_0) + \left[\begin{smallmatrix} I \\ \Gamma^0 \end{smallmatrix}\right]\right) = \left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_1) & -I \end{smallmatrix}\right] \left(F_e(\theta_1) + \left[\begin{smallmatrix} I \\ \Gamma^0 \end{smallmatrix}\right]\right), \tag{4.91}$$

$$\left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_0) & -I \end{smallmatrix}\right] F_r(\theta_0) = \left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_1) & -I \end{smallmatrix}\right] F_r(\theta_1). \tag{4.92}$$

In these equations $F_e(\theta_0) = 0$ and $F_r(\theta_0) = 0$, such that the combination is

$$\left[\begin{smallmatrix} I & 0 \\ 0 & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_1) & -I \end{smallmatrix}\right] [F_e(\theta_1) + \left[\begin{smallmatrix} I \\ \Gamma^0 \end{smallmatrix}\right] \;\; F_r(\theta_1)]. \tag{4.93}$$

When this equation is pre-multiplied with $(I - G(\theta_1))^{-1} \check{H}(q, \theta_1) \left[\begin{smallmatrix} I & 0 \\ \Gamma(\theta_1) & -I \end{smallmatrix}\right]$ on both sides, and then $\left[\begin{smallmatrix} 0 & (I - G(\theta_1))^{-1} R(q,\theta_1) \end{smallmatrix}\right]$ is added on both sides, it is obtained that

$$T(q, \theta_0) = T(q, \theta_1), \tag{4.94}$$

By condition *(c)* the model set is globally network identifiable at θ_0 such that

$$T(\theta_0) = T(\theta_1) \Rightarrow M(\theta_0) = M(\theta_1). \tag{4.95}$$

■

4.7.4 Proof of Theorem 4.8

First the proof of part *1* is given. The pdf of the innovation $\check{e}$ is given by 2 equations: there is the normal distribution of $e = [I \;\; 0]\check{e}$

$$f(e) = \frac{(2\pi)^{-\frac{p}{2}}}{|\Lambda|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} e^T \Lambda^{-1} e\right), \tag{4.96}$$

and

$$\begin{bmatrix} \Gamma^0 & -I \end{bmatrix} \check{e} = 0 \text{ w.p. } 1. \tag{4.97}$$

The likelihood for N datapoints is then also given by 2 equations (Srivastava and von Rosen, 2002; Khatri, 1968)

$$L_a(\theta) = \frac{(2\pi)^{-\frac{pN}{2}}}{|\Lambda(\theta)|^{\frac{N}{2}}} \exp\left(-\frac{1}{2} \varepsilon_a^T(t, \theta) \Lambda^{-1}(\theta) \varepsilon_a(t, \theta)\right), \tag{4.98}$$

and

$$\begin{bmatrix} \Gamma(\theta) & -I \end{bmatrix} \varepsilon(t, \theta) = 0 \text{ w.p. } 1 \quad \forall t. \tag{4.99}$$

Then taking the natural logarithm results in

$$\begin{aligned} \log L_a(\theta) =& c - \frac{N}{2} \log\det \Lambda(\theta) \\ & - \frac{1}{2} \sum_{t=1}^{N} \varepsilon_a^T(t, \theta) \Lambda^{-1}(\theta) \varepsilon_a(t, \theta). \end{aligned} \tag{4.100}$$

$\log L_a(\theta)$ is the criterion to be maximized combined with (4.99)

$$\begin{aligned} \theta_N^{ML} &= \arg\max_{\theta} \log L_a(\theta) \\ &\text{subject to } 0 = \varepsilon_b(t,\theta) - \Gamma(\theta)\varepsilon_a(t,\theta) \quad \forall t. \end{aligned} \tag{4.101}$$

Taking the sum of squares for each time t gives the equivalent constraint

$$\text{subject to } \frac{1}{N}\sum_{t=1}^{N} Z^T(t,\theta)Z(t,\theta) = 0, \tag{4.102}$$

with Z defined by (4.19).

Now part *2* is proven in a similar way as the maximum likelihood proof in (Åström, 1980) for full rank noise. Under the condition that $\Lambda(\theta)$ and $\varepsilon(\theta)$ do not share parameters, the cost function $\log L(\theta)$ is maximized at

$$\Lambda(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_a(t,\theta)\varepsilon_a^T(t,\theta) \tag{4.103}$$

In this maximum the constraint of (4.24) is satisfied. Then (4.103) is substituted into the objective of (4.24), and added as additional constraint, to obtain (4.26). ■

4.7.5 Proof of Lemma 4.10

The constraint is satisfied when $\Pi(\theta - \theta^*) = 0$ holds. When substituting (4.49) then we have

$$\Pi(S\rho + C - \theta^*) = 0, \tag{4.104}$$

where we have $\Pi S\rho = 0$, such that the constraint is independent of ρ. Substituting $C = -\Pi^\dagger\Pi\theta^*$ then satisfies the equation. ■

4.7.6 Proof of Proposition 4.11

Proof is by substituting $\theta = S\rho + C$ into the CLS (4.20). Lemma 4.10 shows that this parameter mapping satisfies the constraint for all ρ, and thus can be removed. Equivalence of the cost function is trivial. ■

4.7.7 Proof of Proposition 4.12

With $P_\theta = \mathbb{E}(\theta^\star - \hat{\theta}_N)(\theta^\star - \hat{\theta}_N)^T$ and using the mapping (4.49) we get

$$P_\theta = \mathbb{E}S(\rho^\star - \hat{\rho}_N)(\rho^\star - \hat{\rho}_N)^T S^T, \tag{4.105}$$

such that $P_\theta = SP_\rho S^T$. ■

5 Joint-direct identification of a network with algebraic loops

5.1 Introduction

From Chapter 3 we know that a network model can have algebraic loops and still be network identifiable. Since unique network models can be obtained in the presence of algebraic loops, the question that is addressed in this chapter on the basis of (Weerts et al., 2016b) is the following.

> **Under which conditions can consistent estimates be obtained of a dynamic network that contains algebraic loops?**

Treating the situation of algebraic loops is not a typical objective in identification. Some identification methods exist that can handle the presence of algebraic loops in the classical closed-loop system, namely the IV method Söderström and Stoica (1989); Gilson and Van den Hof (2005), and the Two-Stage method Van den Hof and Schrama (1993). For dynamic network versions of these methods see Van den Hof et al. (2013); Dankers (2014). In both these methods the node signals are 'projected' onto an external variable, in order to de-correlate the node signals from the noise. This approach leads to consistent estimates under the presence of algebraic loops, even when process noises are correlated. The price of projection methods is that any excitation due to process noise is de-correlated such that it no longer provides information, and therefore the estimator does not have minimum variance properties.

External excitation must be available for these methods to work, and this excitation must be of 'sufficient' power and order of persistence of excitation.

In order to make an improvement in terms of variance compared to the projection methods, information provided by by process noise can be used in an estimator. The joint-direct method introduced in the previous chapter utilizes the information provided by process noise to achieve Maximum Likelihood estimates of minimum variance, and therefore is an interesting candidate for use in the algebraic loop situation. In the previous chapter we have specified the joint-direct method for networks with strictly proper modules, but in this chapter we formulate the joint-direct method for networks that contain algebraic loops.

Extending the joint-direct method is done by the following approach. The predictor used in Chapter 4 uses strictly past values of node signals to predict the current node values. For networks where feedthrough is present in modules the predictor definition deserves a critical look. We evaluate the predictor for the feedthrough case with and without algebraic loops. In order to avoid the algebraic loop problems, the identification setup will make explicit use of external excitations. The external excitation is only necessary for estimation of the direct feedthrough terms of the modules. Excitation provided by process noise is modeled and used to estimate module dynamics, and so helps to minimize variance.

As an illustrative base-case for developing our approach to the general problem we utilize the symmetric closed-loop system defined in Figure 5.1. This system is a network consisting of the variables $w(t) = \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$, $r(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix}$, $e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}$, and the filters

$$G(q) = \begin{bmatrix} 0 & G_{12}(q) \\ G_{21}(q) & 0 \end{bmatrix}, \quad H(q) = \begin{bmatrix} H_1(q) & 0 \\ 0 & H_2(q) \end{bmatrix}, \quad R(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This closed-loop system is similar to the classical loop used in the Joint-IO method Caines and Chan (1975), however now additional external excitations r_1, r_2 are present. Although the reasoning is built upon the 2-node network presented above, the theory is developed for general dynamic networks that satisfy the following assumptions.

> **Assumption 5.1.** *The data generating system $\mathcal{S}$ is represented by the network model M^0 as defined in Chapter 2 where*
>
> - *the modules in $G^0(q)$ may contain algebraic loops;*
> - *all nodes w are measured;*
> - *some external excitations are present;*
> - *the noise spectrum $\Phi_v(\omega)$ may be non-diagonal and is of rank L.*

This chapter proceeds by evaluating the predictor in the presence of algebraic loops in Section 5.2. Then the identification setup and conditions for consistent estimates are presented in Section 5.3. Obtained results are then discussed in Section 5.4 after which simulations are presented to validate and illustrate the results in Section 5.5.

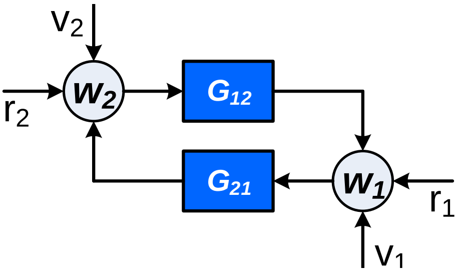


Figure 5.1: Symmetrical closed-loop system $\mathcal{S}_1$.

5.2 Predictor and innovation

5.2.1 Traditional predictor

When there is a direct feedthrough term present in $G_{21}(q)$ the traditional one-step-ahead predictor for w_2 as defined in (2.42) is typically chosen as

$$\breve{w}_2(t|t-1) := \mathbb{E}\{w_2(t) \mid w_2^{t-1}, w_1^t, r_2^t\}, \tag{5.1}$$

where $w_1^t := \{w_1(0), w_1(1), \cdots, w_1(t)\}$, $w_2^{t-1} := \{w_2(0), w_2(1), \cdots, w_2(t-1)\}$ and $r_2^t := \{r_2(0), r_2(1), \cdots, r_2(t)\}$. This predictor will explain the current value of node w_2 based on the past of w_2, and the past and present of node w_1 and external excitation r_2.

In an open-loop situation ($G_{12} = 0$) or in absence of algebraic loops ($G_{12}^\infty = 0$), the $w_1(t)$ is uncorrelated with $e_2(t)$, and the predictor, in terms of transfer functions from the data generating system, is given by

$$\breve{w}_2(t|t-1) = \Big(1 - H_2^{-1}(q)\Big) w_2(t) + H_2^{-1}(q) G_{21}(q) w_1(t), \tag{5.2}$$

such that the innovation is

$$e_2(t) = w_2(t) - \breve{w}_2(t|t-1). \tag{5.3}$$

Note that through the direct feedthrough term G_{21}^∞, the predictor (5.2) will be dependent/conditioned on $w_1(t)$, and not just on the strict past $w_1(t-1)^-$. If $w_1(t)$ is correlated with $e_2(t)$ through a feedthrough term G_{12}^∞, then the prediction of $w_2(t)$ is conditioned onto $e_2(t)$. In that case the innovation is not $e_2(t)$, and the predictor (5.1) is not given by (5.2) anymore. With predictor $\breve{w}_2(t|t-1)$ the noise $e_2(t)$ can be partially predicted from the data, but how much can be predicted depends on the individual variances of e_1 and e_2. When a loop does not contain delay, then the predictor expression belonging of $\breve{w}_2(t|t-1)$ explicitly contains the variances of e_1 and e_2, and is less attractive to use as (5.2). We can conclude that the traditional predictor $\breve{w}_2(t|t-1)$ works fine in open-loop ($G_{12} = 0$) and closed-loop with delay ($G_{12}^\infty G_{21}^\infty = 0$), but becomes difficult to use when an algebraic loop is present.

5.2.2 Traditional network predictor when there are no algebraic loops

The analysis in the section above does not imply that the predictor $\breve{w}_2(t|t-1)$ has no use. In fact the traditional predictor is defined in (2.42) for the MISO estimation problem in dynamic network for the situation that there are no algebraic loops. The MISO estimation setup presented in Chapter 2 may even be extended to a MIMO estimation setup. For a dynamic network, a vector of predictors based on the predictor defined in (2.42) is

$$\breve{w}(t|t-1) := \begin{bmatrix} \breve{w}_1(t|t-1) \\ \vdots \\ \breve{w}_L(t|t-1) \end{bmatrix}, \tag{5.4}$$

where

$$\breve{w}_j(t|t-1) = \mathbb{E}\{w_j(t) \mid w_k^{t-1}, \forall k \in \mathcal{K}_j, w_\ell^t, \forall \ell \in \mathcal{L}_j, r^t\}, \tag{5.5}$$

where $\mathcal{K}_j = \{k \mid G_{jk}^\infty = 0\}$ and $\mathcal{L}_j = \{\ell \mid G_{j\ell}^\infty \neq 0\}$. The expression in terms of network dynamics that belongs to $\breve{w}(t|t-1)$ is obtained in the following proposition for the situation that there are no algebraic loops.

Proposition 5.2. *For a dynamic network considered in* (2.4) *that has no algebraic loops and where* Λ *is diagonal, the vector of one-step-ahead predictors of the node signals* $w(t)$ *is given by*

$$\breve{w}(t|t-1) = W_w^0(q)w(t) + W_r^0(q)r(t), \tag{5.6}$$

with the predictor filters

$$W_w^0(q) = I - (H^0(q))^{-1}(I - G^0(q)), \tag{5.7}$$
$$W_r^0(q) = (H^0(q))^{-1}R^0(q). \tag{5.8}$$

Proof. Provided in Appendix 5.7.1. ■

The obtained predictor expression (5.6) is similar to the predictor expression (4.2) for the situation of full rank noise, however W_w is no longer strictly proper due to the G^∞.

In (2.42) it was established that $\breve{w}_j(t|t-1)$ is uncorrelated to $e_j(t)$ when there are no algebraic loops. It is possible that $G_{12}^\infty \neq 0$, such that $\breve{w}_1(t|t-1)$ becomes correlated to $e_2(t)$. In other words, for the predictor $\breve{w}(t|t-1)$ and innovation $e(t)$ we have the correlation

$$\mathbb{E}\ \breve{w}(t|t-1)e^T(t) \tag{5.9}$$

with 0 on the diagonal, and possibly non-zero values on off-diagonal elements. The vector of predictors $\breve{w}(t|t-1)$ and vector of innovations $e(t)$ are not orthogonal, and so they are not a predictor and innovation pair. This is the reason that $\breve{w}(t|t-1)$ is named a 'vector of predictors', instead of a predictor.

It was shown in Section 2.3 that consistent estimates can be obtained in a multi-input-single-output setting. Since $\breve{w}(t|t-1)$ and $e(t)$ are not a predictor and innovation pair, the standard prediction error framework is not applicable when the vector of predictors $\breve{w}(t|t-1)$ is used in a MIMO estimation setting. In the remainder of the chapter a different definition of the predictor is utilized, which does match the prediction error framework. The focus will be on networks that may have algebraic loops, and this has the situation of no algebraic loops as a special case.

5.2.3 Network predictor

Instead of a predictor which uses instantaneous values of $w(t)$, the predictor in Definition 4.3, namely $\hat{w}(t|t-1) = \mathbb{E}\{w(t) \mid w^{t-1}, r^t\}$, is used to replace the traditional predictor. The predictor is only conditioned with respect to the delayed values w^{t-1}. From the variables that are known at time t, namely w^t, r^t, only $w(t)$ may be correlated with $e(t)$, since e is a white noise process such that $w(t-1)$ and samples before it can not be correlated with $e(t)$. By not conditioning $\hat{w}(t|t-1)$ onto $w(t)$ we prevent the possibility that $\hat{w}(t|t-1)$ is correlated with $e(t)$.

It seems strange for a network which contains modules with feedthrough terms to not condition the predictor onto $w(t)$ since it seems like we discard useful information that is present in $w(t)$. However the effect is that the feedthrough terms are modeled in a different way in the predictor expression. The feedthrough terms appear as part of the noise model.

Proposition 5.3. *For a dynamic network considered in* (2.4) *that satisfied Assumption 5.1, the network predictor* $\hat{w}(t|t-1) = \mathbb{E}\{w(t) \mid w^{t-1}, r^t\}$ *is given by (omitting arguments q, t)*

$$\hat{w}(t|t-1) = \Big(I - (I-G^\infty)^{-1}H^{-1}(I-G)\Big)w + (I-G^\infty)^{-1}H^{-1}Rr. \tag{5.10}$$

Proof. Provided in Appendix 5.7.2. ■

The predictor expression in (5.10) has some interesting properties. For strictly proper modules $G^\infty = 0$, and (5.10) becomes equal to the predictor expression obtained in the previous chapter (4.2) for full rank noise. Through the factor $(I-G^\infty)^{-1}$, the filter $(I-G^\infty)^{-1}H^{-1}(I-G)$ becomes monic, which implies that the predictor filter of w is strictly proper. This conforms to the conditioning on $w(t-1)^-$ that is used in the predictor definition (4.3).

The innovation related to the network predictor $\hat{w}$ is

$$\hat{e}(t) := w(t) - \hat{w}(t|t-1), \tag{5.11}$$

which is equal to a scaled version of the driving noise process

$$\hat{e}(t) = (I-G^\infty)^{-1}e(t). \tag{5.12}$$

Due to the scaling of $e(t)$ the innovation $\hat{e}(t)$ can be correlated over the channels even when $e(t)$ is not.

In the prediction error framework we parameterize a predictor with a model set in order to identify the model, and this leads to consistency under some conditions. Here we can parameterize the predictor (5.10) with a network model, and then evaluate the conditions for consistency. This approach is taken in the next section.

5.3 Joint-direct identification setup

The joint-direct identification setup will be defined for the case of algebraic loops. First the network predictor $\hat{w}$ will be parameterized, and then an identification criterion is defined. The model structure $\mathcal{M}$ is as defined in Chapter 2, and the models in $\mathcal{M}$ satisfy Assumption 5.1. With the use of the parameterized model set the parameterized predictor is defined as

$$\hat{w}(t|t-1;\theta) := W_w(q,\theta)w(t) + W_r(q,\theta)r(t) \tag{5.13}$$

with

$$\begin{aligned} W_w(q,\theta) &= I-(I-G^{\infty}(\theta))^{-1}H^{-1}(q,\theta)(I-G(q,\theta)) \\ W_r(q,\theta) &= (I-G^{\infty}(\theta))^{-1}H^{-1}(q,\theta)R(q,\theta). \end{aligned} \tag{5.14}$$

The parameters of G^{∞} appear as the feedthrough terms of G. From the W_w filter the feedthrough terms of G can not be uniquely determined, this is because W_w is strictly proper, i.e.

$$W_w^{\infty} = I-(I-G^{\infty})^{-1}I(I-G^{\infty}) = 0. \tag{5.15}$$

But since the predictor filters W_w and W_r are uniquely related to the open-loop response T_{wr} and T_{we}, we can uniquely recover a network model from the predictor filters when the network model set is network identifiable. This implies that filter W_r must be used in order to obtain G^{∞}.

The prediction error is defined as $\hat{\varepsilon}(t,\theta) := w(t) - \hat{w}(t|t-1;\theta)$ such that we obtain

$$\hat{\varepsilon}(t,\theta) = (I-G^{\infty}(\theta))^{-1}\varepsilon(t,\theta), \tag{5.16}$$

with

$$\varepsilon(t,\theta) = H^{-1}(q,\theta)\Big((I-G(q,\theta))w(t) - R(q,\theta)r(t)\Big). \tag{5.17}$$

In Chapter 4 ε was used as the prediction error. With this new predictor we use $\hat{\varepsilon}$, being a scaled version of ε.

As identification criterion a weighted least squares criterion will be applied:

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta), \tag{5.18a}$$

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\hat{\varepsilon}^T(t,\theta)Q\hat{\varepsilon}(t,\theta), \tag{5.18b}$$

where the matrix $Q > 0$ is chosen by the user. If the 'true system' is in the model set and indicated by θ_0, then for $\theta = \theta_0$ the prediction error is a white noise. More precisely, for $\theta = \theta_0$ the prediction error is the innovation

$$\hat{\varepsilon}(t,\theta_0) = \hat{e}(t) = (I-G_{\infty})^{-1}e(t). \tag{5.19}$$

In order to reduce the variance of the estimator an appropriate choice for Q is the covariance matrix of $\hat{e}(t)$ being given by $(I-G^{\infty})^{-1}\Lambda(I-G^{\infty})^{-T}$. This will be further commented upon in Section 5.4.

In classical direct and joint-io methods the presence of external signals r is not strictly necessary for arriving at consistency. For dynamic network models that have algebraic loops it has been shown in Chapter 3 that some external excitations r are necessary to have a network identifiable model set. In our new setup the presence of some signals r is necessary for network identifiability, and then also for informativity in view of the presence of direct feedthrough terms in the models. The final step in this section is to prove consistency of the estimator $\hat{\theta}_N$.

> **Theorem 5.4.** *Consider data generated by a system that satisfies Assumption 5.1, and consider a model set $\mathcal{M}$. Let $\hat{\theta}_N$ be defined by (5.18). $M(\hat{\theta}_N)$ is a consistent estimate of M^0 under the following conditions:*
>
> 1. *The network system is in the model set, i.e. $\exists\, \theta_0 \in \Theta$ such that $G(q,\theta_0) = G(q)$, $H(q,\theta_0) = H(q)$, and $R(q,\theta_0) = R(q)$;*
> 2. *The data is informative with respect to $\mathcal{M}$;*
> 3. *The model set $\mathcal{M}$ is globally network identifiable.*

Proof. Provided in Appendix 5.7.3. ■

The conditions of the above theorem are very similar to the conditions of Proposition 4.6. Similarities are due to the fact that the predictor expression (4.2) for full rank networks is a special case of the predictor expression for algebraic loops (5.10). This implies that the joint-direct identification method for full rank networks that was presented in Chapter 4 is a special case of the joint-direct identification method that has been presented in the current chapter. The main difference between the conditions for consistency are the conditions under which the model set is network identifiable. When there are algebraic loops, then some external variables can make the feedthrough terms identifiable.

5.4 Discussion

The identification setup that we have chosen is basically a direct identification method that apparently can estimate a dynamic network / closed-loop system while algebraic loops are present. The basic step that we have made in this respect is to exclude direct feedthrough terms in the predictor models from the node variables w, but we include the direct feedthrough terms in the predictor models from the external signals r. As a result, when algebraic loops are present, the presence of an external excitation signal is necessary. However different from the alternative projection methods (IV, two-stage), we keep on using the full signals w as predictor inputs, rather than projecting them onto external signals first. This has two consequences:

- Firstly, the requirements on the persistence of excitation properties of the external signals will be limited, as the r signals only serve to identify the direct feedthrough terms;

- Secondly, the variance of the estimated models will be driven by the signal power of the w signals, rather than by their projections onto r, thus substantially improving the variance of the estimate.

Since external excitation is used for identification of the direct terms, one could wonder whether it would be necessary to have external excitation signals on all node variables when only a few loops are algebraic. This question can be answered by the conditions under which the model set is network identifiable. For an in-depth reasoning on network identifiability see Chapter 3. Here we make an identifiability analysis for a typical closed-loop system with just one reference, i.e. the system in Figure 5.1 but without r_2. The main question is whether G_{12}^{∞} and G_{21}^{∞} can be determined from T_{wr}^{∞}. To this end the conditions of Proposition 3.9 can be checked. Alternatively we can manually check for identifiability of the feedthrough terms by applying the same reasoning as in Example 3.4, then we attempt to determine the parameters from

$$\begin{bmatrix} 1 & -G_{12}^{\infty}(\theta) \\ -G_{21}^{\infty}(\theta) & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{1-G_{12}^{\infty}G_{21}^{\infty}} \\ \frac{G_{21}^{\infty}}{1-G_{12}^{\infty}G_{21}^{\infty}} \end{bmatrix}}_{T_{wr}^{\infty}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{5.20}$$

It is directly clear that indeed the feedthrough terms can be determined uniquely. This implies that for the typical closed-loop system just one external excitation is necessary to deal with an algebraic loop.

One could wonder whether it would be necessary to apply the presented network predictor to all node variables. Without addressing this problem in detail, it seems feasible to use the traditional predictor for predicting those node variables that are a-priori known to have no algebraic loops, while applying the new network predictor for the variables that are part of an algebraic loop. For this approach the nodes that are part of an algebraic loops need to be predicted jointly, while the other nodes may be predicted separately. When in the network node w_i is not part of an algebraic loop, and noise v_i is independent of other noises, then in the parameterized predictor (5.13) the parameters on the row corresponding to w_i are independent of parameters on the other rows. When a row has independent parameters, then the estimation problem of that particular row may be separated from estimation of the other parameters. This means that in order to estimate the network we only need to jointly predict and estimate the nodes where parameters are shared.

Concerning the asymptotic variance of the estimate, it can be stated that minimum variance is achieved when the covariance of the innovations process $\hat{e}(t)$ is used as weighting Q in the identification criterion. Now suppose the covariance of innovation $\hat{e}(t)$ is parameterized in the model set $\mathcal{M}$ as $\hat{\Lambda}(\theta)$. According to Ljung (1999) the resulting asymptotic (minimum) variance is equal to the asymptotic variance of the maximum likelihood estimator under Gaussian assumptions, resulting in the criterion

$$V_N(\theta) = \frac{1}{N} \det \sum_{t=1}^{N} \hat{\varepsilon}(t,\theta)\hat{\varepsilon}^T(t,\theta). \tag{5.21}$$

This criterion has the property that no weight has to be chosen in order to obtain minimum variance estimates.

5.5 Simulation

The direct identification method with network predictor $\hat{w}$ will be validated by numerical simulations. A comparison to the extended instrumental variable method Gilson and Van den Hof (2005) is made since it is one of the methods that can deal with algebraic loops. $\mathcal{S}_1$ is used to generate data, and has the following dynamical components:

$$\begin{aligned} G_{12}(q) &= 0.3 + 0.7q^{-1} + 0.3q^{-2}, \\ G_{21}(q) &= 0.15 + 0.9q^{-1} - 0.5q^{-2}, \\ H_1(q) &= 1, \\ H_2(q) &= 1, \\ R(q) &= I. \end{aligned}$$

Low order FIR filters are used to keep numerical computation relatively easy.

In total two sets of experiments are performed. For each set of experiments 100 Monte-Carlo random simulations are performed. The external excitation is known but generated as normally distributed white noise $r_i = \mathcal{N}(0, \sigma_r^2)$, and randomized in each experiment. The first set of experiments is performed with the power of the noise $(e_i = \mathcal{N}(0, \sigma_e^2))$, and the power of the external excitation equal, $\sigma_e^2 = \sigma_r^2 = 1$. In the second set of experiments the external excitation has less power to illustrate the benefit of the additional excitation coming from the noise, $\sigma_e^2 = 1, \sigma_r^2 = 0.01$. For all experiments $N = 1000$ data samples are drawn, and initial conditions are 0.

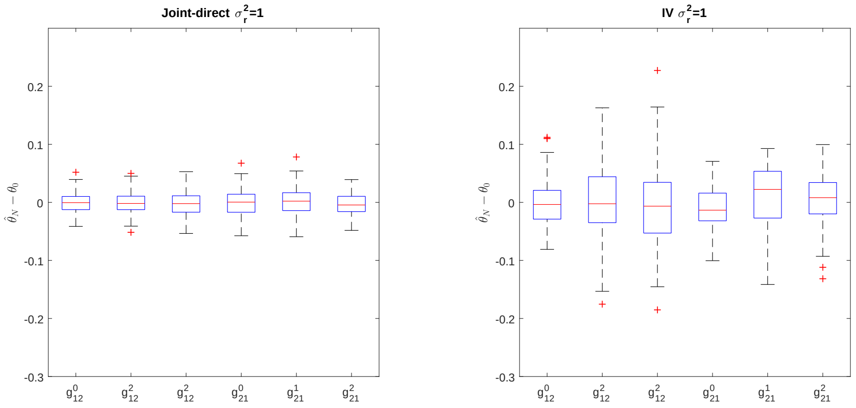


Figure 5.2: Difference between estimated and true parameters over 100 experiments with $\sigma_r^2 = \sigma_e^2$ for the two estimation methods, Joint-direct in the left plot, IV in the right plot.

For the identification methods the following setup is used. The prediction error $\hat{\varepsilon}(t, \theta)$ is considered with the criterion defined in (5.18), and the model set is chosen to contain the network system. For weighting matrix Q we choose the optimal weighting

$$\Lambda = \mathbb{E}\hat{e}_0(t)\hat{e}_0^T(t).$$

This optimal weight is unknown in practice, but in such a situation the determinant minimization criterion (5.21) can be used, leading to the same asymptotic variance Ljung (1999). The cost function (5.18) is minimized in Matlab by the function `fmincon()` without imposing constraints.

The estimated models are evaluated by plotting the difference between estimated parameters and the true parameters as a boxplot. For the situation that power of noise and excitation is equal the results are plotted in Figure 5.2. It is clear that both methods deliver estimates around the true parameter values. The main difference is that the IV estimates appear to have a higher variance.

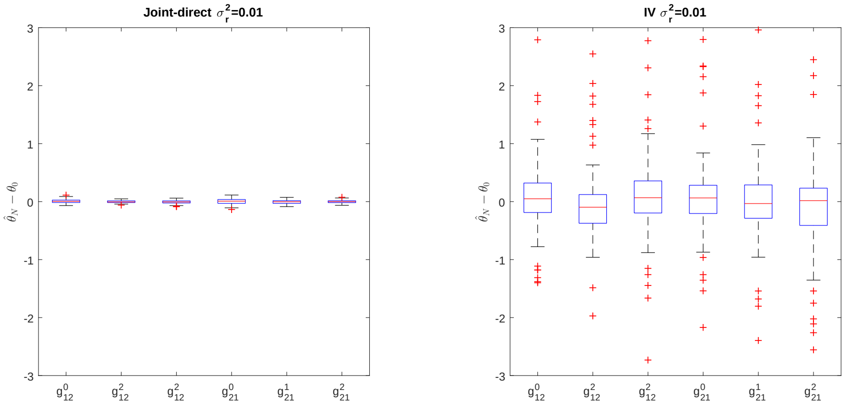


Figure 5.3: Difference between estimated and true parameters over 100 experiments with $\sigma_r^2 = 0.01\sigma_e^2$ for the two estimation methods, Joint-direct in the left plot, IV in the right plot.

It has been mentioned that the joint-direct method does not require large power of r since excitation by process noise is used. This situation is plotted in Figure 5.3. Here the estimates are also around the true parameter values and both methods appear unbiased. However now the difference in variance between the two methods is much larger, the joint-direct method significantly outperforms the IV. This difference is caused by the fact that the IV depends on the external excitation, but that signal has too low power.

5.6 Conclusions

The joint-direct method can obtain consistent estimates of dynamic networks that contain algebraic loops. Differences in conditions compared to the strictly proper modules situation is due to requirements on external excitations. Variance of estimated networks and requirements on external excitations are reduced compared to instrumental variable methods. The drawback of the method is that a non-convex optimization problem is to be solved, while with IV it is possible to use a convex optimization problem.

5.7 Appendix

5.7.1 Proof of Proposition 5.2

The dynamic network (2.4) can be written as

$$w = \big[I - H^{-1}(I-G)\big]\, w + H^{-1}Rr + e. \tag{5.22}$$

When applying the definition of (5.4) to w there are 3 terms to evaluate, which we do for each w_j

$$\begin{aligned}&\breve{w}_j(t|t-1) = \\ &\mathbb{E}\left\{ w_j(t) - [H^{-1}]_{j\star}(I-G)w(t) \;\middle|\; w(t-1)^-, w_k(t) \text{ if } G^{\infty}_{jk} \neq 0, r(t)^- \right\} \cdot \\ &\cdot + \mathbb{E}\left\{ [H^{-1}]_{j\star}Rr \;\middle|\; w(t-1)^-, w_k(t) \text{ if } G^{\infty}_{jk} \neq 0, r(t)^- \right\} \cdot \\ &\cdot + \mathbb{E}\left\{ e_j \;\middle|\; w(t-1)^-, w_k(t) \text{ if } G^{\infty}_{jk} \neq 0, r(t)^- \right\}.\end{aligned} \tag{5.23}$$

In the first term, the direct terms that appear are exactly the ones where $G^{\infty}_{jk} \neq 0$, such that the conditional expectation results in $w_j(t) - [H^{-1}]_{j\star}(I-G)w(t)$. In the second term, the $r(t)$ and its past are fully present in the conditioning, such that the expectation results in $[H^{-1}]_{j\star}Rr$. For the third term it must be evaluated whether $e_j(t)$ is a part of the $w_k(t)$ corresponding to $G^{\infty}_{jk} \neq 0$. Whenever $G^{\infty}_{jk} \neq 0$, then the condition on not having algebraic loops implies that every path from w_j to w_k has a delay, such that $w_k(t)$ can not be correlated with $e_j(t)$. This implies that the third term is 0. Then stacking every predictor together the predictor expression (5.6) is obtained. ■

5.7.2 Proof of Proposition 5.3

In the proof, arguments q and t will be omitted where possible. Starting with (2.4) add a multiplication with identity after H, and subtract w from both sides of the equation:

$$0 = -(I-G)w + Rr + H(I-G^{\infty})(I-G^{\infty})^{-1}e. \tag{5.24}$$

Pre-multiplying the equation with $(I-G^{\infty})^{-1}H^{-1}$ and adding w to both sides of the equation delivers

$$\begin{aligned} w =& \big\{I - (I-G^{\infty})^{-1}H^{-1}(I-G)\big\}w + \\ &+ (I-G^{\infty})^{-1}H^{-1}Rr + (I-G^{\infty})^{-1}e. \end{aligned} \tag{5.25}$$

Since the first filter on the right hand side is strictly proper, the second filter is proper, and e is white noise, it follows directly, by applying the definition of the network predictor, that the predictor is given by the first two terms on the right hand side. ■

5.7.3 Proof of Theorem 5.4

The proof is divided into 3 parts. The first part is the convergence of $V_N(\theta)$ to $\bar{V}(\theta) := \bar{\mathbb{E}}\, \hat{\varepsilon}^T(t,\theta)\Lambda^{-1}\hat{\varepsilon}(t,\theta)$ for $N \to \infty$. This convergence can be shown by applying the convergence proof found in Ljung (1999).

As second part of the proof it is shown that the true system minimizes the quadratic function $\bar{V}(\theta)$. Rewrite the prediction error in terms of its driving variables with the use of $w = (I-G)^{-1}(He + Rr)$ (omitting arguments q, t)

$$\begin{aligned}\hat{\varepsilon}(\theta) = &\big(I - G_\infty(\theta)\big)^{-1}H^{-1}(\theta)\big(I-G(\theta)\big)(I-G)^{-1}H^{-1}e + \cdot\\ &\cdot + \big(I - G_\infty(\theta)\big)^{-1}H^{-1}(\theta)\Big\{\big(I-G(\theta)\big)(I-G)^{-1}R(q) - R(q,\theta)\Big\}r.\end{aligned} \tag{5.26}$$

The above equation contains a mix of parameterized and non-parameterized transfer functions. In the above equation the e terms can be split into a delayed and non-delayed part

$$\begin{aligned}&\Big\{\big(I - G_\infty(\theta)\big)^{-1}H^{-1}(\theta)\big(I-G(\theta)\big)(I-G)^{-1}H^{-1} +\\ &-\big(I-G_\infty\big)^{-1}\Big\}e(t) + \big(I-G_\infty\big)^{-1}e(t),\end{aligned} \tag{5.27}$$

where the first e term is delayed, and the second non-delayed. The two terms are uncorrelated since $e(t)$ is a white noise. In the quadratic function $\bar{V}(\theta)$ any cross-term between the non-delayed e term and r or the delayed e term is zero, due to uncorrelatedness. The choice of parameter has no effect on the non-delayed and non-parameterized e term, and it has no cross-terms in the expectation of the quadratic expression. The choice $\theta = \theta_0$ results in the first term in (5.27) and the second (r-dependent) term in (5.26) to be equal to 0. This minimizes $\bar{V}(\theta)$ and the prediction error is then equal to the innovation $\varepsilon(t,\theta_0) = \hat{e}(t)$. In the last step we consider any model θ_1 which realizes the same criterion $\bar{V}(\theta)$ as θ_0,

$$\bar{V}(\theta_0) = \bar{V}(\theta_1). \tag{5.28}$$

It can be shown that

$$\begin{aligned}\bar{V}(\theta_1) - \bar{V}(\theta_0) = &\,\bar{\mathbb{E}}\,(\hat{\varepsilon}_1 - \hat{\varepsilon}_0)^T\Lambda^{-1}(\hat{\varepsilon}_1 - \hat{\varepsilon}_0) +\\ &+2\bar{\mathbb{E}}(\hat{\varepsilon}_1 - \hat{\varepsilon}_0)^T\Lambda^{-1}\hat{\varepsilon}_0\end{aligned}$$

where $\hat{\varepsilon}_i := \hat{\varepsilon}(t,\theta_i)$.

Analysing the second term we can use the fact that $\hat{\varepsilon}_0 = e_0(t)$ being a white noise process, while $\hat{\varepsilon}_1 - \hat{\varepsilon}_0$ can be shown to be dependent on data up to $t-1$ only. Therefore this latter term will be uncorrelated with $e(t)$, and the second term in the above equation will be 0, so that

$$\bar{V}(\theta_1) - \bar{V}(\theta_0) = \bar{\mathbb{E}}\,(\hat{\varepsilon}_1 - \hat{\varepsilon}_0)^T\Lambda^{-1}(\hat{\varepsilon}_1 - \hat{\varepsilon}_0). \tag{5.29}$$

With the condition on informativity of data it now follows that $\bar{V}(\theta_0) - \bar{V}(\theta_1) = 0$ implies that the corresponding predictor filters must be equal. Then with the use of network identifiability we know that this must also imply that the models are equal

$$M(\theta_0) = M(\theta_1), \tag{5.30}$$

which concludes the proof. ■

6 Single module identification - input selection[1]

6.1 Introduction

Estimation of a single module is relevant for local identification in a dynamic network, and a relevant question for this topic is the following.

> **Which selection of node signals allows for consistent estimation of a module of interest?**

There are some different approaches in the literature which treat this problem of selecting which inputs to use for estimation of a single module.

- The approach in (Dankers et al., 2016) is based on the idea that unmeasured nodes can be removed from the network, resulting in a new transformed network. Depending on which nodes are measured, the module of interest can be left invariant in the transformed network. When the module of interest is left invariant, then it can be estimated consistently using just those measured nodes in a direct identification method under certain conditions. In particular the process noise of predictor inputs and outputs must be uncorrelated. An extension of the conditions is presented in (Dankers et al., 2017) for some situations of correlated noise. A two-stage method has been proposed in (Dankers et al., 2016) as an alternative to the direct estimation, and this leads to a different set of conditions for consistency. In particular there are no more restrictions on correlatedness of noises, but the requirements on external excitation are increased.

[1] This chapter is based on collaboration with Jonas Linder. A paper based on this joint work is in preparation.

- The approach in (Linder and Enqvist, 2017a,b; Linder, 2017) is also based on the idea that unmeasured nodes can be removed from the network, resulting in a new transformed network. In this approach, even when the module of interest is modified in the transformed network, it may be possible to recover the module of interest from modules in the transformed network. An indirect identification method is used to identify the module of interest.
- The approach in (Bazanella et al., 2017; Hendrickx et al., 2018) is based on an indirect identification approach for the situation that all nodes are excited by external variables. An analysis of the identifiability of the module of interest is made in particular for the information present in the measured nodes.

The approaches in (Dankers et al., 2016) and (Linder and Enqvist, 2017a) are both based on the removal of unmeasured nodes, and the objective in this chapter is to generalize these approaches. Like in previous chapters a-priori assumptions on diagonality of H or R will be avoided. There is no specific identification method chosen, instead identifiability of the module of interest is analyzed such that an identification method of choice can be used. One possible identification method is the joint-direct method, which can lead to significantly smaller variance than the indirect identification approaches in the literature.

It is possible that there are multiple sets of measured nodes that each lead to consistent estimates of a module of interest. However each of these sets of nodes can result in different variance for the estimated module of interest. The question which set of nodes leads to the lowest variance is interesting, but not addressed in this chapter.

The approach in (Dankers et al., 2016) is based on the immersion algorithm, which applies the lifiting technique. Lifting a path essentially works as follows. When there are modules connecting $w_1 \to w_2 \to w_3$, and w_2 is unmeasured, then module G_{32} is removed and replaced by a new module $\breve{G}_{31} = G_{32}G_{21}$, such that the path then no longer passes through node w_2. In the immersion algorithm, each path that passes through an unmeasured node is lifted, and afterwards the unmeasured nodes are deleted from the network. The core mechanism of lifting paths is the following: Due to the lifting of paths, unmeasured nodes no longer influence the measured nodes, while the behavior of measured nodes remains the same. Immersion is illustrated by an example.

Example 6.1. *Consider the network in Figure 6.1 where the nodes are described by the following equations*

$$\begin{aligned} w_1 &= & G_{12}w_2 + G_{13}w_3 + G_{14}w_4 + r_1 + v_1 \\ w_2 &= & G_{24}w_4 + r_2 + v_2 \\ w_3 &= & r_3 + v_3 \\ w_4 &= G_{41}w_1 & + r_4 + v_4 \end{aligned} \tag{6.1}$$

where node w_4 is unmeasured and to be immersed. The paths from w_1 through w_4 and the path from external signals v_4 and r_4 need to be lifted. When the lifting technique is applied to path $w_1 \to w_4 \to w_2$ then module G_{24} is replaced by a new module $\breve{G}_{21} = G_{24}G_{41}$, and similar for the external signals. Lifting the path $w_1 \to w_4 \to w_1$ leads to a self-loop around node w_1, i.e. node w_1 is an input to itself

$$w_1 = G_{14}G_{41}w_1 + G_{12}w_2 + G_{13}w_3 + r_1 + v_1, \tag{6.2}$$

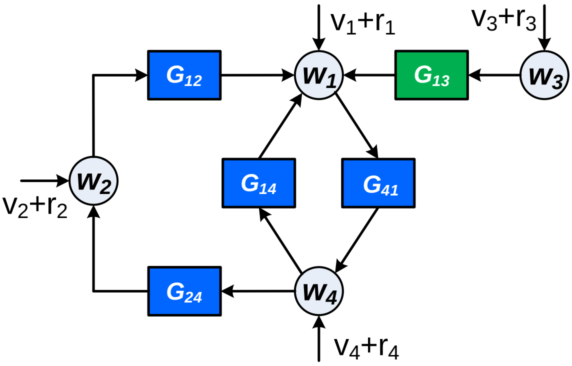


Figure 6.1: A network where node w_4 is non-measured.

which is not allowed in the network setup as defined in Chapter 2. The self-loop can be resolved to obtain a network description that matches our definition, by moving both w_1 terms to the left-hand side, and normalizing by multiplication of the equation with $S = (1 - G_{14}G_{41})^{-1}$. Finally the network depicted in Figure 6.2 is obtained, where the module of interest has changed.

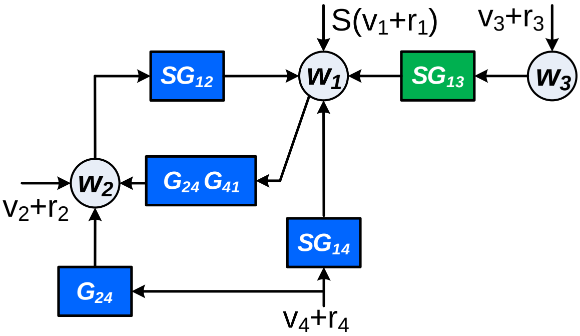


Figure 6.2: Network obtained after immersion of node w_4 of the network depicted in Figure 6.1.

The method of indirect inputs introduced in (Linder and Enqvist, 2017a) is an alternative technique to remove unmeasured nodes. With indirect inputs, the main idea is that the out-neighbor of an unmeasured node contains information about that unmeasured node. Then the equation of the out-neighbor is manipulated in order to obtain an explicit expression for the unmeasured node, which is then used to eliminate the unmeasured node from the network by inserting the equation. An example where the method is applied is shown next.

Example 6.2. *Consider the network in Figure 6.1 with node w_4 unmeasured, and with w_2 the indirect observation of the unmeasured node. There are two modules that* *have w_4 as input that need to be removed. In order to remove G_{14} we make use of the equation of the indirect observation, $w_2 = G_{24}w_4 + v_2 + r_2$ can be transformed into*

$$w_4 = G_{24}^{-1}(w_2 - v_2 - r_2). \tag{6.3}$$

Now the new equation is used to eliminate w_4 as an input to node w_1 by inserting the above equation into the equation of node w_1

$$w_1 = G_{12}w_2 + G_{13}w_3 + G_{14}\underbrace{G_{24}^{-1}(w_2 - r_2 - v_2)}_{w_4} + r_1 + v_1. \tag{6.4}$$

The other module G_{24} can be removed by lifting the path $w_1 \to w_4 \to w_2$. Then the network that is obtained is depicted in Figure 6.3.

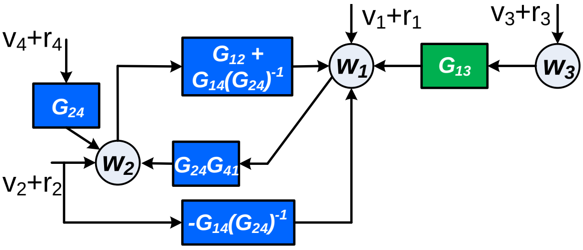


Figure 6.3: Modification of the network depicted in Figure 6.1, obtained after removal of node w_4 by the indirect inputs method.

The major difference that the indirect inputs method makes is that the inverse of modules may appear in the obtained network. Now the implications for identification of the module of interest in the transformed networks obtained by the two node removal approaches are analyzed. The objective is to identify module G_{13} while w_4 is unmeasured. When immersion is used to remove w_4, the network in Figure 6.2 can be used to determine which inputs to use. Node w_1 is the output, and nodes w_2 and w_3 are inputs. Additionally the external variables r_1 and r_4 are in-neighbors of w_1 and are included as predictor inputs. The noise v_4 acts as a confounding variable. In the alternative situation that indirect observation w_2 is used to remove w_4, then the network in Figure 6.3 can be used to determine which inputs to use. Here, also node w_1 is the output, and nodes w_2 and w_3 are inputs. However now the external variables r_1 and r_2 are in-neighbors of w_1 and must be included as predictor inputs. Now the noise v_2 acts as the confounding variable. The implication for identification is that different external variables are used as an input, and that noises must be modeled in a different way. Additionally the identified module will be different depending on which of the two approaches is chosen. Conditions under which the module of interest remains invariant will be discussed later in the chapter.

In the presented examples the module G_{13} is invariant when w_4 is removed with one method, but not with the other method. Immersion and the indirect inputs method each have a different set of conditions for which nodes can be removed such that a module of interest remains invariant. The two approaches described above are generalized in this chapter such that a generalized set of conditions for invariance is obtained.

The approach in this chapter is as follows. Transforming networks and removing nodes is defined as the concept named *abstraction*. It will be shown that immersion and the indirect inputs method are particular ways of abstracting a network, and a method that generalizes the two special cases is presented. For single module identification we may exploit situations where the module of interest remains invariant in the abstracted network, and where the module is identifiable in the abstracted network. It is investigated under which conditions a module remains invariant and identifiable when the generalized method is used to abstract a network. Finally, given a network, it is determined which nodes to measure such that a module remains invariant.

The chapter will continue as follows. In Section 6.2, it is shown that transformations can be applied to a network representation to manipulate the modules and structure. Then the concept of abstraction is formally defined and related to the network transformations. In Section 6.3, particular network abstractions for identification purposes are discussed, in particular immersion, indirect inputs and a generalization of those. Section 6.4 contains a discussion on conditions under which a module of interest is invariant for particular abstraction algorithms. Finally, in Section 6.5, a method to select which nodes to measure such that a local module can be consistently identified is developed.

6.2 Equivalent network representations

In the previous section, the immersion and indirect inputs methods have been introduced that modify paths of a network representation before removing a node. A particular feature for those two methods is that the behavior of the remaining nodes remains invariant while the modules change. The core mechanism that is used is that a particular network can be represented by different network models with different topology. The basic idea of the two discussed methods consists of two parts, first modules are transformed such that unmeasured nodes no longer are an input to measured nodes, and then the unmeasured nodes can be removed without changing the behavior of measured nodes. In this section, we investigate which freedom is available for transforming the dynamics and removing the nodes. First the focus is on transforming the network to alternative representations, and later on removing nodes.

6.2.1 Transformation of the global network

Fundamentally, we need to define when two networks are equivalent descriptions of behavior, and what freedom is available to transform the network to an equivalent one. In the network model definition in Chapter 2, it has been stated that the external variables r and nodes w are known, and it is reasonable to state that equivalent networks must describe the same relation between r and w. The dynamic influence of r on w is described by the open-loop transfer function matrix T_{wr}, and so the equivalence of two networks additionally requires equality of the two related open-loop transfer function matrices from r to w. The open-loop response of the network is described by (2.18), i.e. $w(t) = T_{wr}r(t) + \bar{v}(t)$. When w, r and T_{wr} are the same for two networks, then also $\bar{v}$ must be the same.

Definition 6.3. *Let the network model $M^{(i)}$ correspond with open-loop transfer $T_{wr}^{(i)}$ and noise spectrum $\Phi_{\bar{v}}^{(i)}$ for $i = \{1,2\}$. Network models $M^{(1)}$ and $M^{(2)}$ are said to be equivalent if*

$$T_{wr}^{(1)} = T_{wr}^{(2)} \quad \textit{and} \quad \Phi_{\bar{v}}^{(1)} = \Phi_{\bar{v}}^{(2)}. \tag{6.5}$$

In the above definition the $T_{wr}^{(i})$ and $\Phi_{\bar{v}}^{(i)}$ are associated with w and r for $i = \{1,2\}$. There is an implicit assumption in the definition that w and r are the same for both $i = \{1,2\}$.

The full freedom that is available for transformation of a network model to an equivalent network model is characterized by operations applied to the network equation. For example, the network equation (2.4) can be pre-multiplied by a rational transfer matrix P, i.e.

$$P(q)w(t) = P(q)\Big(G(q)w(t) + R(q)r(t) + v(t)\Big). \tag{6.6}$$

The above pre-multiplication can lead to a left-hand side unequal to $w(t)$, in which case the offending terms must be moved to the right-hand side, i.e.

$$w(t) = (I - P(q))w(t) + P(q)\Big(G(q)w(t) + R(q)r(t) + v(t)\Big), \tag{6.7}$$

which is denoted as

$$w(t) = G^{(2)}(q)w(t) + R^{(2)}(q)r(t) + v^{(2)}(t) \tag{6.8}$$

where

$$G^{(2)} = I - P(I - G), \quad R^{(2)} = PR, \quad v^{(2)} = Pv. \tag{6.9}$$

An interesting feature of the network transformations is that the response from external variables and process noises to internal variables remains the same. A pre-multiplication P as defined above leaves the transfer T_{wr} invariant, i.e.

$$T_{wr} = (P(I - G))^{-1}PR, \tag{6.10}$$

where $P^{-1}P = I$.

We are looking for transformations P that lead to an appropriate network representation, i.e. the transformed network corresponds to a network model as defined in Chapter 2. These restrictions are as follows:

- if P is proper, then $G^{(2)}$ is proper,
- if P is stable, then $G^{(2)}$ is stable,

- and P should be such that $G^{(2)}$ is hollow, i.e. the diagonal of $(I - P(I - G))$ is 0.

In terms of manipulating the network, properness of $G^{(2)}$ is not important. However, in an identification context usually only proper modules will be considered. It is not strictly necessary that P is proper to have a proper $G^{(2)}$, moreover even with non-proper $G^{(2)}$ it might be possible to identify some modules. Despite these arguments, properness of P is assumed to make the reasoning suitable for use with most conventional identification methods. The lifting example from Example 6.1 is continued to show that the corresponding transformation P creates a zero column in $G^{(2)}$.

Example 6.4 (Example 6.1 continued)**.** *In Example 6.1 the immersion of node w_4 has been demonstrated. In the current example the corrsponding transformation matrix P associated with the lifting procedure is evaluated. Figure 6.2 shows the immersed network, and before w_4 is removed the associated input matrix is*

$$R^{(2)} = \begin{bmatrix} S & 0 & 0 & SG_{14} \\ 0 & 1 & 0 & G_{24} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.11}$$

From (6.9) *we find that $P = R^{(2)}$ since $R = I$ in the original network. It can be validated using* (6.9) *that indeed the modules of the transformed network are*

$$\begin{aligned} G^{(2)} &= I - \begin{bmatrix} S & 0 & 0 & SG_{14} \\ 0 & 1 & 0 & G_{24} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -G_{12} & -G_{13} & -G_{14} \\ 0 & 1 & 0 & -G_{24} \\ 0 & 0 & 1 & 0 \\ -G_{41} & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & SG_{12} & SG_{13} & 0 \\ G_{24}G_{41} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ G_{41} & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{6.12}$$

What can be observed is that the transformation is precisely such that in the top-left element of $G^{(2)}$ we create $1 - \frac{1-G_{14}G_{41}}{1-G_{14}G_{41}} = 0$. Moreover the right-most column of $G^{(2)}$ becomes 0 due to the multiplication with P. In this way an equivalent network is created that does not use w_4 as input to any module.

In (6.9) a transformation of the noise model is defined. When we describe the noise model as $v = H^{(1)}e$, then a pre-multiplication with P does not necessarily lead to a monic filter $PH^{(1)}$. For that reason $H^{(2)}$ and $\Lambda^{(2)}$ are obtained through spectral factorization of the transformed noise spectrum

$$P(e^{-j\omega})\Phi_v(\omega)P^T(e^{j\omega}) = H^{(2)}(e^{-j\omega})\Lambda^{(2)}(H^{(2)}(e^{j\omega}))^T. \tag{6.13}$$

It has already been discussed that there are some restrictions on P, but a large freedom in the choice of transformation P is left.

Proposition 6.5. *A network model $M^{(1)}$ containing the modules $G^{(1)}$ can be transformed into an equivalent network model $M^{(2)}$ containing the modules $G^{(2)}$ using the transformation defined by*

$$P = (I - G^{(2)})(I - G^{(1)})^{-1} \tag{6.14}$$

for any $G^{(1)}$ and $G^{(2)}$ that satisfy the dynamic network definition in Chapter 2.

Proof. Provided in Appendix 6.7.1. ■

The consequence of transforming $G^{(1)}$ to an arbitrary $G^{(2)}$ is that the corresponding $R^{(2)}$ will have a complex structure

$$R^{(2)} = (I - G^{(2)})(I - G^{(1)})^{-1}R^{(1)}. \tag{6.15}$$

The implication is that when $G^{(1)}$ is transformed, then $R^{(2)}$ will compensate the changes to keep node behavior invariant. This also holds for the noise model, which will contain additional correlations. Without any further restrictions on the choice of R and H, the modules represented in G contain no information on the dynamic network. It is the combination of G, R, H that determines the dynamic network.

6.2.2 Local module transformation

In this section a particular network transformation is introduced that later on is an important tool to relate the indirect inputs method to immersion. Modules in G can be transformed to arbitrary other transfer functions, but here we investigate what the consequences are of changing a single module. To this end we will change the direction of a module in the network, i.e. a module is 'flipped'. With 'flipping' is meant that a module $G_{ji}^{(1)}$ that connects w_i to w_j is replaced by a module $G_{ij}^{(2)} = (G_{ji}^{(1)})^{-1}$ that connects w_j to w_i. This is illustrated by the following example.

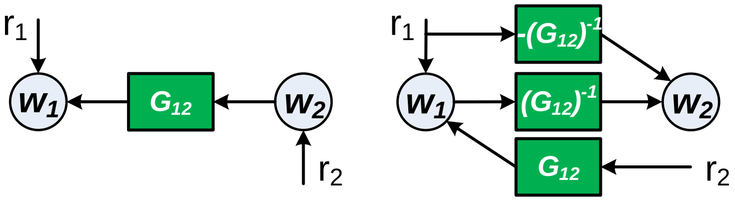
Example 6.6. *Consider the network in Figure 6.4a. If we transform the network to an equivalent one with module G_{12} 'flipped', then we can find a transformation that leads to the desired new $G^{(2)}$. The original network is characterized by*

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 & G_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \tag{6.16}$$

In the transformed network after flipping we have $G^{(2)} = \begin{bmatrix} 0 & 0 \\ G_{12}^{-1} & 0 \end{bmatrix}$, and so the corresponding transformation is determined using Proposition 6.5

$$P = \begin{bmatrix} 1 & 0 \\ -G_{12}^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -G_{12} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & G_{12} \\ -G_{12}^{-1} & 0 \end{bmatrix}. \tag{6.17}$$

For this transformation we have $R^{(2)} = PR^{(1)} = P$, as can be observed in the transformed network in Figure 6.4b. The effect of r_1 and r_2 on the nodes must remain invariant under transformation, so they are essentially re-routed around the flipped



(a) (b)

Figure 6.4: Simple network on the left, and the network with flipped arrow on the right.

module. Excitation r_1 originally does not affect w_2 directly, and with the flipped module its contribution through G_{12}^{-1} must be compensated for by a direct link filtered with $-G_{12}^{-1}$.

6.2.3 Abstraction

The next step is to extend network equivalence with the option to remove nodes from the representation. To this end the concept *abstraction* is defined next.

> **Definition 6.7.** *Let network model $M^{(1)}$ be associated with nodes $w^{(1)} \in \mathbb{R}^L$, external variables $r \in \mathbb{R}^K$, open-loop transfer function $T_{wr}^{(1)} \in \mathbb{R}^{L\times K}$, and noise spectrum $\Phi_{\bar{v}}^{(1)} \in \mathbb{R}^{L\times L}$. Let network model $M^{(2)}$ be associated with nodes $w^{(2)} \in \mathbb{R}^{L_2}$, external variables $r \in \mathbb{R}^K$, open-loop transfer $T_{wr}^{(1)} \in \mathbb{R}^{L_2\times K}$, and noise spectrum $\Phi_{\bar{v}}^{(2)} \in \mathbb{R}^{L_2\times L_2}$. Let C be the matrix that selects $w^{(2)}$ from $w^{(1)}$, so define C with one 1 per row, zeros everywhere else, full row rank, and such that $w^{(2)} = Cw^{(1)}$. Network model $M^{(2)}$ is said to be an abstraction of $M^{(1)}$ if*
>
> $$T_{wr}^{(2)} = CT_{wr}^{(1)}, \quad \Phi_{\bar{v}}^{(2)} = C\Phi_{\bar{v}}^{(1)}C^T. \tag{6.18}$$
>
> *The nodes that are in $w^{(1)}$, but not in $w^{(2)}$ are said to be abstracted from the network.*

When one network is an abstraction of the other network it means that some nodes have been removed from the abstracted network compared to the original network. Moreover the behavior of the nodes in the abstracted network is the same as the behavior of the corresponding nodes in the original network.

The next step is to determine how to obtain an abstraction of a network. In certain cases abstracting nodes $\bar{w}$ from a network can be done by simply pre-multiplying the network equation (2.4) with the selection matrix C, i.e.

$$Cw(t) = C\Big(G(q)w(t) + R(q)r(t) + v(t)\Big). \tag{6.19}$$

However, this only is an abstraction if the abstracted nodes $\bar{w}$ no longer appear on the right-hand side of the equation. If $\bar{w}$ appears on the right-hand side of (6.19) then the abstracted nodes have an influence on the behavior of the nodes in Cw, such that (6.18) cannot hold. It has to be determined how to define a transformation P such that an abstraction can be obtained.

A node w_i influences other nodes through its out-neighbors, and these corresponding modules are a column in G. When a node has no influence on the rest of the network, then it has no out-neighbors, and the corresponding column is 0. Abstracting node w_i requires us to transform the network such that a 0-column is formed by transformation, after which the node can be removed. By Proposition 6.5 we know that such a transformation always exists. Then the abstraction satisfies the relations

$$G^{(2)} = C\left(I - P(I-G)\right)C^T, \quad R^{(2)} = CPR. \tag{6.20}$$

A noise model constructed as $CPH^{(1)}$ is a non-square matrix, which is difficult to handle in an identification setting. Therefore the transformed noise model $H^{(2)}$, $\Lambda^{(2)}$ will be obtained through spectral factorization

$$CP(e^{-j\omega})\Phi_v(\omega)P^T(e^{j\omega})C^T = H^{(2)}(e^{-j\omega})\Lambda^{(2)}(H^{(2)}(e^{j\omega}))^T. \tag{6.21}$$

6.3 Abstraction for identification of a single module

6.3.1 Identification approach

The identification approach taken is to identify the module of interest from the abstracted network. We have seen in the previous section that either the module of interest G_{ji} is invariant under abstraction

$$G_{ji}^{(1)} = G_{ji}^{(2)}, \tag{6.22}$$

or the module has been modified during abstraction

$$G_{ji}^{(1)} \neq G_{ji}^{(2)}. \tag{6.23}$$

We will investigate what is needed in order to consistently identify the module of interest in each of these two situations.

In the situation that the module of interest is invariant, then we can use the identification methods from previous chapters for identification of that module. If we can define a network identifiable network model set that contains the abstracted network, then consistent estimates may be obtained. We have seen in Chapter 3 that a network model set is identifiable when $\begin{bmatrix} R & H \end{bmatrix}$ is parameterized such that it can be given a leading diagonal by column operations. A transformation used to abstract the network may result in a complex structure in $R^{(2)}$, hence a model set containing the abstracted network may not satisfy the diagonality condition. In fact the module of interest may not be identifiable due to the great flexibility required to model the complex structure in $R^{(2)}$, regardless of which conditions are checked.

In the alternative situation that the module of interest has changed as a result of abstraction, then the identification procedure has to take that change into account. An indirect identification method may estimate all the modules of the abstracted network, and then in a second step obtain the module of interest from the estimated abstracted network. Another option is that the abstracted network is parameterized with modules that share parameters, e.g. by assigning parameters to $P(\theta)$ and $G^{(1)}(\theta)$ to obtain the parameterization of $G^{(2)}$. The latter case is difficult to handle in an identification setting due to the shared parameters, and it is not clear under which conditions the module of interest will be identifiable. In both the indirect inputs method, and with the shared parameters, it has to be investigated whether the module of interest is identifiable from the abstracted network model, see for example (Bazanella et al., 2017).

The approach taken in the remainder of the chapter is to consider abstracted networks where the module of interest remains invariant, and for which a network identifiable model can be defined. In particular we investigate abstraction algorithms that lead to abstracted networks for which network identifiable model sets can be defined. When the abstraction algorithms have been defined it will be investigated under which conditions the module of interest remains invariant.

6.3.2 Immersion

The immersion algorithm is formalized as an algorithm to remove one node, which can be applied sequentially for each node to be removed. The order in which nodes are removed is of no consequence for the final result.

Algorithm 6.8 (Immersion of node w_k, (Dankers et al., 2016))**.** *The immersion algorithm to remove node w_k from the network, where w_k has in-neighbors w_p, $p \in \mathcal{P}$, and out-neighbors w_c, $c \in \mathcal{C}$, is defined as follows:*

1. *Noise v_k moves downstream to all out-neighbors w_c, $c \in \mathcal{C}$, where the new noise is*
$$v_c^{new} = v_c + G_{ck} v_k \quad \textit{for all } c \in \mathcal{C}.$$
2. *Excitation u_k moves downstream to all out-neighbors w_c, $c \in \mathcal{C}$, where the new excitation is*
$$u_c^{new} = u_c + G_{ck} u_k, \quad \textit{for all } c \in \mathcal{C}.$$
3. *The dynamics of modules from all in-neighbors w_p, $p \in \mathcal{P}$, to w_k are merged with the dynamics of modules from w_k to all out-neighbors w_c, $c \in \mathcal{C}$, i.e.*
$$G_{cp}^{new} = G_{cp} + G_{ck} G_{kp}, \quad \textit{for all } c \in \mathcal{C}, p \in \mathcal{P}.$$
4. *Self-loops are removed by normalizing nodes that are both in-neighbor and out-neighbor of w_k, i.e. for all j for which $G_{jj}^{new} \neq 0$ set $\breve{G}_{jj} = 0$ and*

compensate

$$G_{ji}^{(2)} = \frac{1}{1 - G_{jk}G_{kj}} G_{ji}^{new}, \qquad \textit{for all } i \in \mathcal{N} \setminus j, \tag{6.24}$$

$$v_j^{(2)} = \frac{1}{1 - G_{jk}G_{kj}} v_j^{new}, \quad u_j^{(2)} = \frac{1}{1 - G_{jk}G_{kj}} u_j^{new}. \tag{6.25}$$

The immersion algorithm can be written using the transformation of a network. Consider the situation that we have a network with $R = I$ denoted as

$$\begin{bmatrix} w_{\mathcal{S}} \\ w_{\mathcal{Z}} \end{bmatrix} = \begin{bmatrix} G_{\mathcal{S}\mathcal{S}} & G_{\mathcal{S}\mathcal{Z}} \\ G_{\mathcal{Z}\mathcal{S}} & G_{\mathcal{Z}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} w_{\mathcal{S}} \\ w_{\mathcal{Z}} \end{bmatrix} + \begin{bmatrix} r_{\mathcal{S}} \\ r_{\mathcal{Z}} \end{bmatrix} + \begin{bmatrix} v_{\mathcal{S}} \\ v_{\mathcal{Z}} \end{bmatrix} \tag{6.26}$$

with $w_{\mathcal{Z}}$ as the unmeasured nodes to be removed by immersion. Immersion can be explained as substituting the equation for $w_{\mathcal{Z}}$ into the equation for $w_{\mathcal{S}}$ to remove the dependence on the unmeasured nodes. Then the transformation that corresponds to applying the immersion algorithm for each unmeasured node is

$$P_{imm} = \begin{bmatrix} D_{\mathcal{S}} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & G_{\mathcal{S}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} \\ 0 & I \end{bmatrix}, \tag{6.27}$$

where $D_{\mathcal{S}}$ is diagonal with $(D_{\mathcal{S}})_{jj} = \frac{1}{1 - G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}j}}$. The transformed network obtained by transformation matrix P_{imm} is as follows:

$$G^{(2)} = I - \begin{bmatrix} D_{\mathcal{S}} & 0 \\ 0 & I \end{bmatrix} \underbrace{\begin{bmatrix} I & G_{\mathcal{S}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} I - G_{\mathcal{S}\mathcal{S}} & -G_{\mathcal{S}\mathcal{Z}} \\ -G_{\mathcal{Z}\mathcal{S}} & I - G_{\mathcal{Z}\mathcal{Z}} \end{bmatrix}}_{\begin{bmatrix} I - G_{\mathcal{S}\mathcal{S}} - G_{\mathcal{S}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}\mathcal{S}} & 0 \\ -G_{\mathcal{Z}\mathcal{S}} & I - G_{\mathcal{Z}\mathcal{Z}} \end{bmatrix}}. \tag{6.28}$$

The effect of $D_{\mathcal{S}}$ is that self-loops are removed, since it contains exactly the inverse of the diagonal of $I - G_{\mathcal{S}\mathcal{S}} - G_{\mathcal{S}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1}G_{\mathcal{Z}\mathcal{S}}$. This transformation has the effect that in the transformed network the nodes no longer depend on the to be abstracted nodes. Then the immersed network is obtained by transforming with P_{imm} and abstracting the nodes $w_{\mathcal{Z}}$.

For the network described in (6.26), the $R^{(2)}$ determined by immersion has the following structure

$$R^{(2)} = \begin{bmatrix} I & 0 \end{bmatrix} P_{imm} = \begin{bmatrix} D_{\mathcal{S}} & D_{\mathcal{S}}G_{\mathcal{S}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} \end{bmatrix} R, \tag{6.29}$$

where $R = I$. Since $D_{\mathcal{S}}$ is diagonal by definition, $R^{(2)}$ has a leading diagonal block by construction. We can conclude that immersion of a network that has diagonal R leads to an immersed network that has a leading diagonal in $R^{(2)}$. This has the effect that a network identifiable model set can be chosen for networks that are abstracted by the immersion algorithm if R is diagonal. Later on in Section 6.4 it will be discussed under which conditions the module of interest remains invariant in the immersed network.

A non-square noise model of the same structure as (6.29) is created by the immersion algorithm. As discussed before, a spectral factorization must be applied to obtain a square noise model. After the spectral factorization, the noise model may have a diagonal structure, or not, depending on which correlations were created by the immersion.

6.3.3 Indirect inputs method

The indirect inputs methods is formalized as an algorithm. Figure 6.5 describes the general network setup for the algorithm.

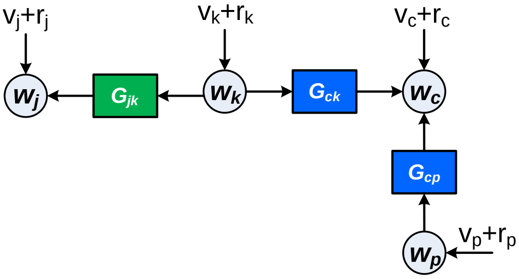


Figure 6.5: Basic network for the application of the indirect inputs algorithm.

Algorithm 6.9 (Removal of w_k using indirect observation w_c)**.** *The indirect inputs algorithm to remove w_k from a network, where w_k has a out-neighbor w_c and where w_c is not an in-neighbor of w_k, is defined on the basis of (Linder and Enqvist, 2017a) as follows: Denote $\mathcal{P}_c$ as the set of in-neighbors of w_c and $\mathcal{P}_k$ as the set of in-neighbors of w_k. Module G_{ck} is assumed to be non-zero.*

1. *The direction of G_{ck} is 'flipped', i.e. G_{ck} is removed, and a new module G_{ck}^{-1} is added with w_c as input and w_k as output.*
2. *Compensate the effect of all the in-neighbors of w_c, except for w_k, i.e.*
   - *Noise v_c forms an additional connection $v_k^{new} = -G_{ck}^{-1}v_c$,*
   - *Excitation u_c forms an additional connection $u_k^{new} = -G_{ck}^{-1}u_c$,*
   - *In-neighbors of w_c except for w_k form an additional connection*

   $$G_{kp}^{new} = G_{ck}^{-1}G_{cp} \text{ for all } p \in \mathcal{P}_c \setminus k.$$
3. *Excitations of w_k are moved to w_c with additional dynamics, i.e.*
   - *Noise v_k is moved $v_c^{new} = v_c + G_{ck}v_k$,*
   - *Excitation u_k is moved $u_c^{new} = u_c + G_{ck}u_k$,*
   - *In-neighbors of w_k re-route the modules*

   $$G_{cp}^{new} = G_{ck}G_{kp} + G_{cp} \text{ for all } p \in \mathcal{P}_k.$$
4. *The other modules and excitations are unaffected.*
5. *Node w_k is immersed using Algorithm 6.8.*

An interesting observation is that steps 1-3 of the algorithm are the same steps as described in Section 6.2.2 where the causal direction of a module was 'flipped'. In the final step, the node is immersed, implying that one way of interpreting the indirect inputs method for removing a single node is a combination of flipping arrows and then immersing the transformed network.

A transformation matrix can be derived that corresponds to steps 1-3. The network depicted in Figure 6.5 is

$$\begin{bmatrix} w_j \\ w_c \\ w_k \\ w_p \end{bmatrix} = \begin{bmatrix} 0 & 0 & G_{jk} & 0 \\ 0 & 0 & G_{ck} & G_{cp} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_j \\ w_c \\ w_k \\ w_p \end{bmatrix} + \begin{bmatrix} r_j \\ r_c \\ r_k \\ r_p \end{bmatrix} + \begin{bmatrix} v_j \\ v_c \\ v_k \\ v_p \end{bmatrix}, \tag{6.30}$$

where it is assumed that w_j and w_c are measured, and w_k and w_p are unmeasured. We can see that the new excitation has the following relation with the old excitation

$$\begin{bmatrix} u_c^{new} \\ u_k^{new} \end{bmatrix} = \begin{bmatrix} 1 & G_{ck} \\ -G_{ck}^{-1} & 0 \end{bmatrix} \begin{bmatrix} u_c \\ u_k \end{bmatrix}. \tag{6.31}$$

This implies that the transformation matrix that achieves step 1-3 is

$$P_{flip} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 1 & G_{ck} & 0 \\ 0 & -G_{ck}^{-1} & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}. \tag{6.32}$$

Now the network identifiability of networks abstracted by Algorithm 6.9 is investigated. The excitation matrix after removing unmeasured nodes is

$$R^{(2)} = \begin{bmatrix} I & 0 \end{bmatrix} P_{imm} P_{flip} R. \tag{6.33}$$

An analysis of the structure of $R^{(2)}$ is made next. From analysis of the immersion it is known that

$$\begin{bmatrix} I & 0 \end{bmatrix} P_{imm} = \begin{bmatrix} I & * \end{bmatrix}, \tag{6.34}$$

i.e. it is structured with a leading diagonal block and arbitrary contents $*$. The bottom left block of P_{flip} contains the non-zero element $-G_{ck}^{-1}$, and this is multiplied by the arbitrary contents of $*$, such that $R^{(2)}$ does not have a leading diagonal. However Theorem 3.10 states that the network is identifiable when $R^{(2)}$ can be given a leading diagonal by column operations. When R is diagonal then it does not affect the structure of $P_{flip}R$, so then in $P_{flip}R$ we can swap the two columns corresponding to w_c and w_k and obtain

$$\bar{P}_{flip} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & G_{ck} & 1 & 0 \\ 0 & 0 & -G_{ck}^{-1} & 0 \\ 0 & 0 & 0 & I \end{bmatrix} R. \tag{6.35}$$

A leading diagonal is created in $\begin{bmatrix} I & * \end{bmatrix} \bar{P}_{flip}$ since the $*$ is multiplied by 0, such that a network abstracted by Algorithm 6.9 has a leading diagonal in $R^{(2)}$. This has the effect that a network identifiable model set can be chosen for networks that are abstracted by the Algorithm 6.9 if there are as many external variables as nodes and R is diagonal.

6.3.4 Generalized algorithm

The immersion and indirect inputs methods are generalized into one algorithm that generates an abstracted network by removing all unmeasured nodes at once, instead of per node. In the indirect inputs method the indirect observation w_c is not allowed to be an in-neighbor of w_j, while in the generalization this restriction is removed. The network is divided into measured nodes $w_{\mathcal{S}}$ and the unmeasured nodes $w_{\mathcal{Z}}$ that are to be abstracted. A combination of the indirect inputs method and immersion can be used for the abstraction. A number of unmeasured nodes that are in-neighbor of w_j are to be indirectly observed through a number of measured nodes, and to this end some notation must be introduced.

- $w_{\mathcal{V}}$ are the unmeasured nodes that are indirectly observed via other nodes, i.e. the nodes are unmeasured but other measured nodes in the network contain information on these nodes;
- $w_{\mathcal{L}}$ are the measured nodes that are indirect input measurements, so they indirectly observe the nodes $w_{\mathcal{V}}$;
- $w_{\tilde{\mathcal{S}}}$ are the remaining measured nodes, i.e. $w_{\tilde{\mathcal{S}}} = w_{\mathcal{S}} \setminus w_{\mathcal{L}}$;
- $w_{\tilde{\mathcal{Z}}}$ are the remaining unmeasured nodes, i.e. $w_{\tilde{\mathcal{Z}}} = w_{\mathcal{Z}} \setminus w_{\mathcal{V}}$.

In the final identification setup the indirect input measurements $w_{\mathcal{L}}$ are used to replace the unmeasured in-neighbors in $w_{\mathcal{V}}$. When the measured and unmeasured nodes are given, then it has to be decided which nodes are considered as indirectly observed nodes, and which are considered as indirect input measurements. In other words, the user has to select the $w_{\mathcal{L}}$ and $w_{\mathcal{V}}$ in a smart way. In the remainder of this section the choice of $w_{\mathcal{L}}$ and $w_{\mathcal{V}}$ is pre-specified, and a systematic way to select these nodes is deferred to a later section.

The network can be rewritten on the basis of the groups of nodes that have been defined above. We have the network that consists of an equation for each of $w_{\tilde{\mathcal{S}}}, w_{\mathcal{L}}, w_{\mathcal{V}}, w_{\tilde{\mathcal{Z}}}$, i.e.

$$\begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \\ w_{\mathcal{V}} \\ w_{\tilde{\mathcal{Z}}} \end{bmatrix} = \begin{bmatrix} G_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}} & G_{\tilde{\mathcal{S}}\mathcal{L}} & G_{\tilde{\mathcal{S}}\mathcal{V}} & G_{\tilde{\mathcal{S}}\tilde{\mathcal{Z}}} \\ G_{\mathcal{L}\tilde{\mathcal{S}}} & G_{\mathcal{L}\mathcal{L}} & G_{\mathcal{L}\mathcal{V}} & G_{\mathcal{L}\tilde{\mathcal{Z}}} \\ G_{\mathcal{V}\tilde{\mathcal{S}}} & G_{\mathcal{V}\mathcal{L}} & G_{\mathcal{V}\mathcal{V}} & G_{\mathcal{V}\tilde{\mathcal{Z}}} \\ G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}} & G_{\tilde{\mathcal{Z}}\mathcal{L}} & G_{\tilde{\mathcal{Z}}\mathcal{V}} & G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \\ w_{\mathcal{V}} \\ w_{\tilde{\mathcal{Z}}} \end{bmatrix} + \begin{bmatrix} u_{\tilde{\mathcal{S}}} \\ u_{\mathcal{L}} \\ u_{\mathcal{V}} \\ u_{\tilde{\mathcal{Z}}} \end{bmatrix} + \begin{bmatrix} v_{\tilde{\mathcal{S}}} \\ v_{\mathcal{L}} \\ v_{\mathcal{V}} \\ v_{\tilde{\mathcal{Z}}} \end{bmatrix}. \tag{6.36}$$

It will be assumed that the indirect input measurements contain sufficient information on the indirectly observed nodes. This means that there are sufficient paths through the network from $w_{\mathcal{V}}$ to $w_{\mathcal{L}}$.

> **Assumption 6.10.** *It is assumed that $G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}$ has full column rank.*

This assumption is on the paths from $w_{\mathcal{V}}$ to $w_{\mathcal{L}}$ contained in $G_{\mathcal{L}\mathcal{V}}$, but also makes use of the paths through the unmeasured nodes in $w_{\tilde{\mathcal{Z}}}$ described by the $G_{\mathcal{L}\tilde{\mathcal{Z}}}(I-G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}$

term. When this transfer function matrix has full column rank, it is implied that the number of indirect input measurements is greater than the number of indirectly observed nodes. The rank condition is generically satisfied when there are $\dim(w_{\mathcal{V}})$ vertex-disjoint paths present from $w_{\mathcal{V}}$ to $w_{\mathcal{L}}$. An example of the full rank assumption is shown in Figure 6.6. In the figure there are the two vertext-disjoint paths $w_{v1} \to w_{l1}$ and $w_{v2} \to w_z \to w_{l3}$ for two nodes that are indirectly observed.

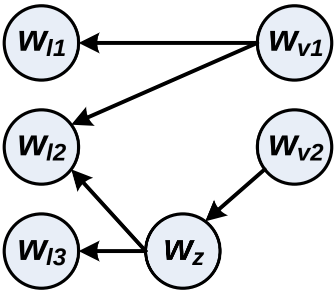


Figure 6.6: Example network with $\mathcal{V} = \{v_1, v_2\}$, $\tilde{\mathcal{Z}} = \{z\}$, $\mathcal{L} = \{l_1, l_2, l_3\}$, where Assumption 6.10 is satisfied.

If just one indirect observation is used to eliminate one unmeasured node, then one module is inverted, resulting in a flipped arrow. In the general situation, paths between indirect input measurements and indirectly observed nodes may pass through a number of other unmeasured nodes, and this has to be taken into account in the abstraction algorithm. Another additional difficulty is that potentially a multivariable transfer has to be inverted, instead of just a single module.

The main idea is that the indirectly observed nodes $w_{\mathcal{V}}$ can be eliminated from the network equation (6.36) by utilizing the equation of the indirect input measurements $w_{\mathcal{L}}$. To this end we need to eliminate the unmeasured nodes $w_{\mathcal{V}}$ and $w_{\tilde{\mathcal{Z}}}$ from the first equation of (6.36). One step is to solve the second equation of (6.36) for $w_{\mathcal{V}}$, and the result into the first equation of (6.36). However it is necessary to appropriately deal with the nodes in $w_{\tilde{\mathcal{Z}}}$. To this end the fourth equation of (6.36) is solved for $w_{\tilde{\mathcal{Z}}}$

$$w_{\tilde{\mathcal{Z}}} = (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} \left(G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}} w_{\tilde{\mathcal{S}}} + G_{\tilde{\mathcal{Z}}\mathcal{L}} w_{\mathcal{L}} + G_{\tilde{\mathcal{Z}}\mathcal{V}} w_{\mathcal{V}} + u_{\tilde{\mathcal{Z}}} + v_{\tilde{\mathcal{Z}}} \right). \tag{6.37}$$

The equation (6.37) is inserted into the second equation of (6.36)

$$\begin{aligned} w_{\mathcal{L}} = {} & \left(G_{\mathcal{L}\tilde{\mathcal{S}}} + G_{\mathcal{L}\tilde{\mathcal{Z}}} (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}} \right) w_{\tilde{\mathcal{S}}} \\ & + \left(G_{\mathcal{L}\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}} (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{L}} \right) w_{\mathcal{L}} \\ & + \left(G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}} (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{V}} \right) w_{\mathcal{V}} \\ & + u_{\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}} (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} u_{\tilde{\mathcal{Z}}} \\ & + v_{\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}} (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} v_{\tilde{\mathcal{Z}}}, \end{aligned} \tag{6.38}$$

and the result is solved for $w_{\mathcal{V}}$

$$\begin{aligned} w_{\mathcal{V}} = \big(G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}\big)^{\dagger}\Bigg(& -\big(G_{\mathcal{L}\tilde{\mathcal{S}}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}}\big)\, w_{\tilde{\mathcal{S}}} \\ & + \big(I - G_{\mathcal{L}\mathcal{L}} - G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{L}}\big)w_{\mathcal{L}} \\ & - u_{\mathcal{L}} - G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}u_{\tilde{\mathcal{Z}}} \\ & - v_{\mathcal{L}} - G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}v_{\tilde{\mathcal{Z}}}\Bigg), \end{aligned} \tag{6.39}$$

where $(.)^{\dagger}$ denotes a left-inverse of a matrix. Then (6.37) is inserted into the first equation of (6.36), after which (6.39) is inserted. In this way the unmeasured in-neighbors in $w_{\tilde{\mathcal{S}}}$ have been replaced by measured nodes.

An important observation on this elimination of equations is the following. The network after it has been manipulated is equivalent to the original network since all changes are due to equation manipulations.

The algorithm that achieves the generalized abstraction is not as simple as just writing Algorithm 6.9 with multivariable signals. There are three main steps:

1. First the paths from all nodes to indirect input measurements $w_{\mathcal{L}}$ that go through other unmeasured nodes $w_{\tilde{\mathcal{Z}}}$ are lifted. This is a multivariable version of the lifting procedure that was described in Example 6.1. It is then ensured that the indirectly observed nodes $w_{\mathcal{V}}$ are in-neighbors to the indirect input measurements $w_{\mathcal{L}}$ like was done in (6.37) and (6.39).
2. The new modules from indirectly observed nodes $w_{\mathcal{V}}$ to input measurements $w_{\mathcal{L}}$ are inverted.
3. All unmeasured nodes are immersed.

Algorithm 6.11 (Generalized algorithm)**.** *The generalized algorithm to remove unmeasured nodes $w_{\mathcal{V}}$ and $w_{\tilde{\mathcal{Z}}}$ from a network representation with use of indirect observations $w_{\mathcal{L}}$. It is assumed that $w_{\mathcal{V}}$, $w_{\tilde{\mathcal{Z}}}$, and $w_{\mathcal{L}}$ are such that Assumption 6.10 is satisfied.*

1. *The paths from all nodes to indirect input measurements $w_{\mathcal{L}}$ that go through other unmeasured nodes $w_{\tilde{\mathcal{Z}}}$ are lifted.*
   - *First the modules $G_{\mathcal{L}\tilde{\mathcal{Z}}}$ are removed by setting*

     $$G^{(2)}_{\mathcal{L}\tilde{\mathcal{Z}}} = 0.$$

   - *Then in order to compensate for the removed modules, the modules* *that have $w_{\mathcal{L}}$ as output need to compensate for the change, i.e.*

$$\begin{aligned} G^{(2)}_{\mathcal{L}\tilde{\mathcal{S}}} &= G_{\mathcal{L}\tilde{\mathcal{S}}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}}, \\ G^{(2)}_{\mathcal{L}\mathcal{L}} &= G_{\mathcal{L}\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{L}}, \\ G^{(2)}_{\mathcal{L}\mathcal{V}} &= G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{V}}. \end{aligned}$$

- *Finally the external signals are compensated*

$$\begin{aligned} u^{(2)}_{\mathcal{L}} &= u_{\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} u_{\tilde{\mathcal{Z}}}, \\ v^{(2)}_{\mathcal{L}} &= v_{\mathcal{L}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} v_{\tilde{\mathcal{Z}}}. \end{aligned}$$

2. *The second objective is to invert the direction of the modules that connect indirectly observed nodes to indirect input measurements.*

- *The direction of $G_{\mathcal{L}\mathcal{V}}$ is flipped, i.e.*

$$G^{(3)}_{\mathcal{V}\mathcal{L}} = (G^{(2)}_{\mathcal{L}\mathcal{V}})^{\dagger}(I - G^{(2)}_{\mathcal{L}\mathcal{L}}).$$

- *The effect that $w_{\mathcal{L}}$ has on itself remains unchanged*

$$G^{(3)}_{\mathcal{L}\mathcal{L}} = G^{(2)}_{\mathcal{L}\mathcal{L}}.$$

Nodes $w_{\mathcal{V}}$ do not depend on themselves as the contribution $G_{\mathcal{V}\mathcal{V}}$ is routed via nodes $w_{\mathcal{L}}$

$$\begin{aligned} G^{(3)}_{\mathcal{V}\mathcal{V}} &= 0, \\ G^{(3)}_{\mathcal{L}\mathcal{V}} &= G^{(2)}_{\mathcal{L}\mathcal{V}} G_{\mathcal{V}\mathcal{V}}. \end{aligned}$$

- *Compensate for the in-neighbors of $w_{\mathcal{L}}$, i.e.*

$$\begin{aligned} v^{(3)}_{\mathcal{V}} &= -(G^{(2)}_{\mathcal{L}\mathcal{V}})^{\dagger} v_{\mathcal{L}}, \\ u^{(3)}_{\mathcal{V}} &= -(G^{(2)}_{\mathcal{L}\mathcal{V}})^{\dagger} u_{\mathcal{L}}, \\ G^{(3)}_{\mathcal{V}\tilde{\mathcal{S}}} &= -(G^{(2)}_{\mathcal{L}\mathcal{V}})^{\dagger} G^{(2)}_{\mathcal{L}\tilde{\mathcal{S}}}, \\ G^{(3)}_{\mathcal{V}\tilde{\mathcal{Z}}} &= -(G^{(2)}_{\mathcal{L}\mathcal{V}})^{\dagger} G^{(2)}_{\mathcal{L}\tilde{\mathcal{Z}}}. \end{aligned}$$

- *In-neighbors of $w_{\mathcal{V}}$ are moved to $w_{\mathcal{L}}$ with additional dynamics, i.e.*

$$\begin{aligned} v^{(3)}_{\mathcal{L}} &= v_{\mathcal{L}} + G^{(2)}_{\mathcal{L}\mathcal{V}} v^{(2)}_{\mathcal{V}}, \\ u^{(3)}_{\mathcal{L}} &= u_{\mathcal{L}} + G^{(2)}_{\mathcal{L}\mathcal{V}} u^{(2)}_{\mathcal{V}}, \\ G^{(3)}_{\mathcal{L}\tilde{\mathcal{S}}} &= G^{(2)}_{\mathcal{L}\tilde{\mathcal{S}}} + G^{(2)}_{\mathcal{L}\mathcal{V}} G_{\mathcal{V}\tilde{\mathcal{S}}}, \\ G^{(3)}_{\mathcal{L}\tilde{\mathcal{Z}}} &= G_{\mathcal{L}\tilde{\mathcal{Z}}} + G^{(2)}_{\mathcal{L}\mathcal{V}} G_{\mathcal{V}\tilde{\mathcal{Z}}}. \end{aligned}$$

3. *All the unmeasured nodes $w_{\tilde{\mathcal{Z}}}$ are immersed.*

- *The modules and signals that are not changed are relabeled*

$$\begin{aligned} v^{(3)}_{\tilde{\mathcal{S}}} &= v_{\tilde{\mathcal{S}}}, \quad v^{(3)}_{\tilde{\mathcal{Z}}} = v_{\tilde{\mathcal{Z}}}, \\ u^{(3)}_{\tilde{\mathcal{S}}} &= u_{\tilde{\mathcal{S}}}, \quad u^{(3)}_{\tilde{\mathcal{Z}}} = u_{\tilde{\mathcal{Z}}}, \\ G^{(3)}_{\tilde{\mathcal{S}}\star} &= G_{\tilde{\mathcal{S}}\star}, \quad G^{(3)}_{\tilde{\mathcal{Z}}\star} = G_{\tilde{\mathcal{Z}}\star}. \end{aligned}$$

- *Define*

$$G^{(3)}_{\mathcal{S}\mathcal{S}} = \begin{bmatrix} G^{(3)}_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}} & G^{(3)}_{\tilde{\mathcal{S}}\mathcal{L}} \\ G^{(3)}_{\mathcal{L}\tilde{\mathcal{S}}} & G^{(3)}_{\mathcal{L}\mathcal{L}} \end{bmatrix}, \quad G^{(3)}_{\mathcal{S}\mathcal{Z}} = \begin{bmatrix} G^{(3)}_{\tilde{\mathcal{S}}\mathcal{V}} & G^{(3)}_{\tilde{\mathcal{S}}\tilde{\mathcal{Z}}} \\ G^{(3)}_{\mathcal{L}\mathcal{V}} & G^{(3)}_{\mathcal{L}\tilde{\mathcal{Z}}} \end{bmatrix},$$

$$G^{(3)}_{\mathcal{Z}\mathcal{S}} = \begin{bmatrix} G^{(3)}_{\mathcal{V}\tilde{\mathcal{S}}} & G^{(3)}_{\mathcal{V}\mathcal{L}} \\ G^{(3)}_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}} & G^{(3)}_{\tilde{\mathcal{Z}}\mathcal{L}} \end{bmatrix}, \quad G^{(3)}_{\mathcal{Z}\mathcal{Z}} = \begin{bmatrix} G^{(3)}_{\mathcal{V}\mathcal{V}} & G^{(3)}_{\mathcal{V}\tilde{\mathcal{Z}}} \\ G^{(3)}_{\tilde{\mathcal{Z}}\mathcal{V}} & G^{(3)}_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix}.$$

Then the final modules of G are

$$G^{(4)} = D\left(G^{(3)}_{\mathcal{S}\mathcal{S}} + G^{(3)}_{\mathcal{S}\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}\mathcal{S}}\right),$$

where $D_{jj} = \frac{1}{1 - G^{(3)}_{j\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}j}}$.

- *The external signals are compensated as well. Define*

$$v^{(3)}_{\mathcal{S}} = \begin{bmatrix} v_{\tilde{\mathcal{S}}} \\ v^{(3)}_{\mathcal{L}} \end{bmatrix}, \quad v^{(3)}_{\mathcal{Z}} = \begin{bmatrix} v^{(3)}_{\mathcal{V}} \\ v_{\tilde{\mathcal{Z}}} \end{bmatrix},$$

$$u^{(3)}_{\mathcal{S}} = \begin{bmatrix} u_{\tilde{\mathcal{S}}} \\ u^{(3)}_{\mathcal{L}} \end{bmatrix}, \quad u^{(3)}_{\mathcal{Z}} = \begin{bmatrix} u^{(3)}_{\mathcal{V}} \\ u_{\tilde{\mathcal{Z}}} \end{bmatrix}.$$

Then after immersion the external signals are

$$\begin{aligned} v^{(4)}_{\mathcal{S}} &= D\left(v^{(3)}_{\mathcal{S}} + G^{(3)}_{\mathcal{S}\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}v^{(3)}_{\mathcal{Z}}\right), \\ u^{(4)}_{\mathcal{S}} &= D\left(u^{(3)}_{\mathcal{S}} + G^{(3)}_{\mathcal{S}\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}u^{(3)}_{\mathcal{Z}}\right), \end{aligned}$$

where D is defined above.

When $G^{(3)}_{\mathcal{V}\mathcal{L}}$ is defined, the $G^{(2)}_{\mathcal{L}\mathcal{L}}$ is the compensation for interconnections that exist between indirect observations $w_{\mathcal{L}}$, which is 0 in the situation that just 1 node is removed by indirect inputs.

The algorithm corresponds with a transformation defined by the matrix

$$P = \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & G^{(3)}_{\mathcal{S}\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1} \\ 0 & I \end{bmatrix} P^{flip}, \tag{6.40}$$

with D and $G^{(3)}$ as defined in Algorithm 6.11 and

$$P^{flip} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & G^{(2)}_{\mathcal{LV}} & 0 \\ 0 & -\left(G^{(2)}_{\mathcal{LV}}\right)^{\dagger} & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}.$$

Similar to the identifiability reasoning in the previous section, when $G^{(2)}_{\mathcal{LV}}$ is diagonal, then $R^{(4)}$ associated with $u^{(4)}_{\mathcal{S}} = R^{(4)}r$ can be given a leading diagonal by column operations. A network identifiable model set may be chosen for the networks abstracted by Algorithm 6.11 if R is diaognal. However in case $G^{(2)}_{\mathcal{LV}}$ is non-diagonal then there is no leading diagonal in $R^{(4)}$. This does not necessarily make the network non-identifiable, as the leading diagonal condition is just sufficient. In such a situation a more thorough analysis of identifiability must be performed before conclusions can be drawn. The identifiability result is formalized in the following proposition.

> **Proposition 6.12.** *Consider a network model M where the nodes are grouped as in* (6.36)*, where R and $G_{\mathcal{LV}}$ are both square and diagonal, and where $G_{\mathcal{L}\tilde{\mathcal{Z}}} = 0$. The $R^{(4)}$ of the abstracted network model $M^{(4)}$ obtained by Algorithm 6.11 can be given a leading diagonal by column operations, such that a globally network identifiable model set can be chosen that contains the abstracted network model.*

Proof. Provided in Appendix 6.7.3. ■

Since the immersion and indirect inputs methods have been generalized by Algorithm 6.11, the above proposition also shows identifiability for the immersion and indirect inputs methods. This identifiability result uses the sufficient condition of Theorem 3.10, which is the reason for requiring diagonality of R and $G_{\mathcal{LV}}$. On the basis of the discussion in Chapter 3 we can conclude that the above proposition can be generalized quite a lot. In future research we may evaluate identifiability conditions that are also necessary, and we may evaluate identifiability of just the module of interest. Moreover further research into topological conditions that guarantee identifiability of abstracted networks can be made. It should be noted that any transformation P can be applied before immersion, and some of those may lead to network representations for which a network identifiable model set can be defined. This means that there are other algorithms that can obtain abstracted networks that are suitable for identification.

6.4 Invariance of a module

In the previous section, an algorithm has been shown to generate an abstracted network representation that leads to network identifiable model sets. When additionally the module of interest G_{ji} remains invariant in the abstracted network, then the nodes of the abstracted network may be used to obtain consistent estimation of this module. For the situation that Algorithm 6.11 is used to generate an abstracted network representation, the question to be addressed is under which conditions the module of interest is invariant.

As was shown in Example 6.1 and 6.2, modules might remain invariant. Both for the immersion and the indirect inputs methods there are sets of conditions that specify when a module remains invariant under transformation. In this section, these conditions will be generalized into one set of conditions such that a module of interest remains invariant for the new Algorithm 6.11.

If we have a network, and we chose the sets of nodes to be used in Algorithm 6.11, then it will depend on the original network topology whether the module of interest is affected. There are two structures of the original network that lead to changes in the module of interest in the immersion process (Dankers et al., 2016), and these may lead to problems in the abstraction Algorithm 6.11. The following two examples illustrate these two issues. Noise-free networks are used in the examples in order to stick to the core reasoning.

Example 6.13 (Parallel paths)**.** *Consider the module of interest G_{ji}. Paths that run in parallel to this module, i.e. paths from w_i to w_j, may lead to changes in the module of interest during abstraction. Consider the network in Figure 6.7a. If w_u is removed using immersion, then the path $w_i \to w_u \to w_j$ is lifted and the dynamics of modules G_{ju}, G_{ui} are merged with module of interest $w_i \to w_j$, i.e.*

$$w_j = (G_{ji} + G_{ju}G_{ui})w_i. \tag{6.41}$$

The typical way to prevent these parallel paths from changing the module of interest is by retaining a node in every parallel path in the abstracted network, for example by measuring w_u. An alternative is to include w_l as an indirect measurement of w_u. Also in that case an equation is used that does not contain w_i, and G_{ji} remains invariant, i.e.

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}w_l, \tag{6.42}$$

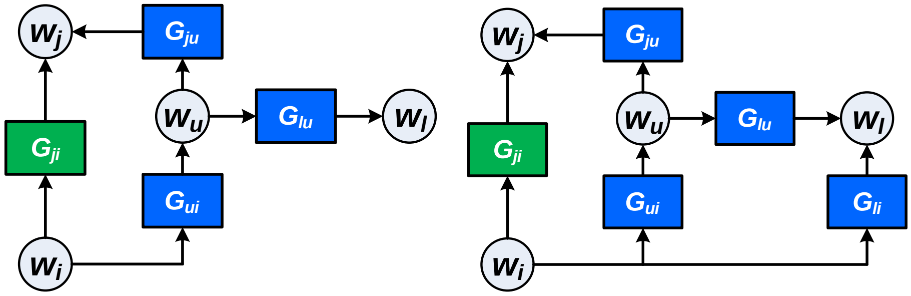
where $w_u = G_{lu}^{-1}w_l$ is used. When an additional path $w_i \to w_l$ exists as in the network in Figure 6.7b, then the situation changes. Now the equation for node w_l depends on w_i, and if the unknown node w_u is eliminated using the equation for w_l, then an additional contribution from w_i appears such that the module of interest is changed, i.e.

$$w_j = (G_{ji} - G_{ju}G_{lu}^{-1}G_{li})w_i + G_{ju}G_{lu}^{-1}w_l, \tag{6.43}$$

where $w_u = G_{lu}^{-1}(w_l - G_{li}w_i)$ is used. If in Figure 6.7a there is no path from w_u to w_l, then w_l cannot be used as an indirect input measurement.

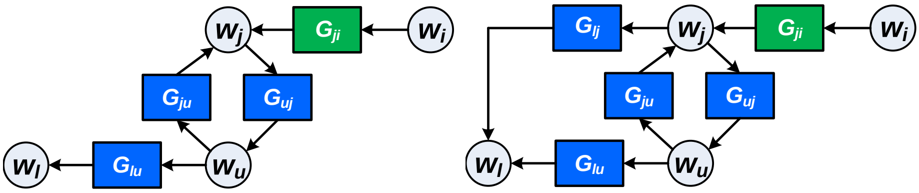
From the example it can be observed that the nodes used as indirect input measurements should not have w_i as an in-neighbor.

Example 6.14 (Self-loops)**.** *Suppose the module of interest is G_{ji}. Paths that run as a loop around the output of this module, i.e. paths from w_j to w_j, may lead to changes in the module of interest during abstraction. Consider the network in Figure 6.8a. If the node w_u of the network in Figure 6.8a is eliminated by immersion, then a self-loop from w_j to itself is created. A way to prevent these self-loops is by retaining a node in every loop around the output w_j in the abstracted network.*



(a) (b)

Figure 6.7: Networks to illustrate issues with parallel paths when abstracting.



(a) (b)

Figure 6.8: Networks to indicate issues with self-loops when making abstractions.

An alternative is to have w_u eliminated by its out-neighbor w_l, in which case w_u is substituted for an equation that does not depend on w_j. This alternative does not create a self-loop either, i.e.

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}w_l, \tag{6.44}$$

with $w_u = G_{lu}^{-1}w_l$. If instead there is a direct link $w_j \rightarrow w_l$ like in Figure 6.8b, then w_l depends directly on w_j, and using this equation for elimination of w_u would again lead to a dependence of w_j on itself in the abstracted network, i.e.

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}(w_l - G_{lj}w_j), \tag{6.45}$$

with $w_u = G_{lu}^{-1}(w_l - G_{lj}w_j)$. The self-loop should be normalized, leading to

$$w_j = \frac{G_{ji}}{1 + G_{ju}G_{lu}^{-1}G_{lj}}w_i + \frac{G_{ju}G_{lu}^{-1}}{1 + G_{ju}G_{lu}^{-1}G_{lj}}w_l \tag{6.46}$$

where it is obvious that the module of interest has changed.

In conclusion, it is not sufficient to only consider parallel paths from w_i to w_j and self-loops from w_j to w_j that appear in the data generating system. Paths from w_i and w_j to the indirect observations $w_{\mathcal{L}}$ also have to be considered to avoid merging of paths and to keep G_{ji} invariant under the transformation. These observations are formalized in the conditions below.

Proposition 6.15. *Consider a dynamic network model associated with modules $G^{(1)}$ where the nodes are divided into the groups $w_{\tilde{\mathcal{S}}}, w_{\mathcal{L}}, w_{\mathcal{V}}, w_{\tilde{\mathcal{Z}}}$. Assume that nodes $w_{\mathcal{L}}$ and $w_{\mathcal{V}}$ have been chosen such that Assumption 6.10 is satisfied. Consider the abstracted network model associated with modules $G^{(2)}$ obtained by Algorithm 6.11. Define the sets $\mathcal{J} = \{j\} \cup \mathcal{L}$ and $\mathcal{D} = \mathcal{V} \cup \tilde{\mathcal{S}} \backslash j$. The module of interest G_{ji} remains invariant under abstraction, i.e. $G_{ji}^{(1)} = G_{ji}^{(2)}$, if the following conditions are satisfied:*

(a) All paths from w_i to $w_{\mathcal{J}}$, excluding the direct path G_{ji}, pass through a node $w_k, k \in \mathcal{D} \setminus i$,

(b) All paths from w_j to $w_{\mathcal{J}}$ pass through a node $w_k, k \in \mathcal{D}$.

Proof. Provided in Appendix 6.7.2. ■

In condition *(a)* the index i is excluded from the set $\mathcal{D}$ for the following reason. Every path from w_i starts in w_i, so all those paths contain a node in $\mathcal{D}$.

The set $\mathcal{D}$ is the set of, either directly or indirectly, observed nodes except for node j. This means that all parallel paths from w_i to $w_{\mathcal{J}}$ must pass through a node that is observed either directly or indirectly. Similarly all loops around the 'output', i.e. all paths from w_j to $w_{\mathcal{J}}$ must pass through a node that is observed either directly or indirectly.

Remark 6.16. *The conditions in Proposition 6.15 are a generalization of the conditions for immersion. When the sets $\mathcal{L} = \emptyset$ and $\mathcal{V} = \emptyset$ are chosen, then Algorithm 6.11 is equivalent to the immersion algorithm when applied to all unmeasured nodes. Moreover the conditions for invariance of the module of interest that have been specified in (Dankers et al., 2016) are obtained. With the new conditions, parallel paths $w_i \to w_j$ and loops around the output $w_j \to w_j$ can also be blocked by indirectly observed nodes, instead of just by directly observed nodes.*

Remark 6.17. *The conditions in Proposition 6.15 are a generalization of the conditions for the indirect inputs method. When it is assumed that indirect observations are not an in-neighbor of the output node, i.e. $G_{jl} = 0, l \in \mathcal{L}$, and when the remaining unmeasured nodes are assumed to not be an in-neighbor of indirect input measurements, i.e. $0 = G_{ln}, l \in \mathcal{L}, n \in \tilde{\mathcal{Z}}$, then Algorithm 6.11 is equivalent to the indirect inputs method (Linder and Enqvist, 2017a). Moreover the conditions for invariance of the module of interest that have been specified in (Linder and Enqvist, 2017a) are obtained. With the new conditions the indirect input measurements $w_{\mathcal{L}}$ can be an in-neighbor of w_j in the original network, and that indirect input measurements may have unmeasured nodes $w_{\mathcal{Z}}$ as an in-neighbor.*

If we have selected as set of nodes that are measured, and it is decided which nodes are indirectly observed by the chosen indirect input measurements, then it can be checked whether the module of interest remains invariant in the abstracted network. Since a network identifiable model set can be selected for the abstracted network, and the module of interest is invariant, we satisfy important conditions for consistent estimation of the target module. A suitable identification method is to be chosen, and this depends for example on whether noises have become correlated in the abstracted network.

6.5 Choosing measured nodes

Proposition 6.15 allows us to check whether a module remains invariant under abstraction if the network topology is known and we have divided the nodes into four groups. The next question is how to choose the sets of nodes, based on the network topology, such that the module of interest remains invariant.

The strategy to obtaining a set of measured nodes in (Dankers et al., 2016) is as follows. First the input and output nodes of the module of interest are required to be measured. Then every parallel path from the input to the output node must be blocked by a measured node. This means that nodes are added such that each of those paths contains a measured node. Similarly every loop around the output node must be blocked by a measured node, so nodes are added such that each of those loops contains a measured node. Different nodes on a path can be chosen to block the path, so the choice of which nodes to measure is not unique.

Now with the extension to the possibility of having indirectly observed nodes the method of choosing nodes is adapted. A parallel path or a loop can now be blocked by either a measured or an indirectly observed node. However when we use an indirect observation to block a path, then additional conditions must be satisfied. Paths from either input or output of the module of interest to the indirect observation must also be blocked by either a measured or an indirectly observed node. For each indirect observation that is added, this condition on blocking the paths is applied recursively. This selection method is demonstrated in the following example.

Example 6.18 (Selecting nodes). *Consider the network in Figure 6.9. It is illustrated how to select nodes such that the conditions of Proposition 6.15 are satisfied. The module of interest is G_{ji}, so we select w_j as output, and w_i is included as a predictor input. A parallel path through node w_u exists and must be blocked if G_{ji} is to be kept invariant. We can either include w_u as a predictor input, or we can choose to indirectly observe it using w_l. When w_l is chosen as indirect input measurement, then $l \in \mathcal{L}$, and the parallel path from w_i to w_l through w_2 should be blocked, so either w_2 should then be included as a predictor input, or w_3 can be included as the indirect observation of w_2.*

In an estimation setting, both the nodes $w_k, k \in \tilde{\mathcal{S}}\backslash j$ and $w_l, l \in \mathcal{L}$ are used as predictor inputs to parameterized modules. So when identification is performed, what is the difference between having nodes in either of these sets? We have seen that immersion and the indirect inputs method lead to different transformation matrices P and those then lead to different ways that external variables appear in the abstracted network. Particularly the location of zeros in the transformed R is different for immersion and the indirect inputs method. This leads to the need to include different external variables depending on which nodes are in $\tilde{\mathcal{S}}$ and $\mathcal{L}$. The external variables that are

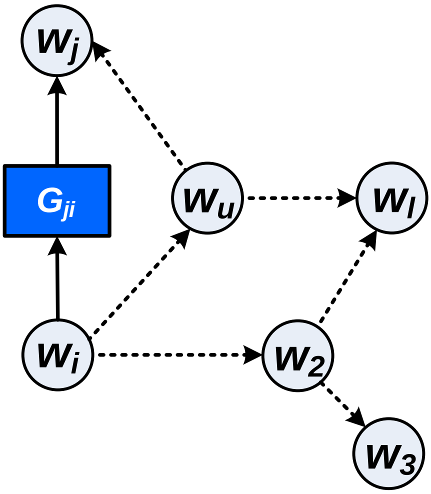


Figure 6.9: Network where measured nodes and indirectly observed nodes are to be selected. Dotted lines are paths through the network.

in-neighbors of a node in the abstracted network are determined in Algorithm 6.11, which are the following: After application of the abstraction algorithm the transformed $u^{(4)}$ has the following relation with the original u

$$u_{\mathcal{S}}^{(4)} = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D\left(G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}\right) & * \end{bmatrix} R \begin{bmatrix} r_{\tilde{\mathcal{S}}} \\ r_{\mathcal{L}} \\ r_{\mathcal{V}} \\ r_{\tilde{\mathcal{Z}}} \end{bmatrix}, \tag{6.47}$$

where $*$ is written for transfer function matrices that are not relevant to the current discussion. When we consider the situation of Proposition 6.12, where R and $G_{\mathcal{L}\mathcal{V}}$ are diagonal, where $G_{\mathcal{L}\tilde{\mathcal{Z}}} = 0$, and considering that D is diagonal, then the following excitations must be included as predictor input:

- For nodes $w_k, k \in \tilde{\mathcal{S}}$ the r_k, $r_{\mathcal{L}}$ and $r_{\tilde{\mathcal{Z}}}$ are in-neighbors and should be included as predictor inputs;
- For nodes $w_l, l \in \mathcal{L}$ the r_l, $r_{\mathcal{L}}$ and $r_{\tilde{\mathcal{Z}}}$ are in-neighbors and should be included as predictor inputs.

In terms of choosing a network model set for the abstracted network, the structure of (6.47) specifies how to choose the zero-structure of the parameterized $R(q, \theta)$ that is to be used for estimation of the abstracted network.

Due to the abstraction the noise process is modified in the same way that the external input process is modified. A similar expression for the abstracted process noise is obtained as for the external variables

$$v_{\mathcal{S}}^{(4)} = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D\left(G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}\right) & * \end{bmatrix} H \begin{bmatrix} e_{\tilde{\mathcal{S}}} \\ e_{\mathcal{L}} \\ e_{\mathcal{V}} \\ e_{\tilde{\mathcal{Z}}} \end{bmatrix}. \tag{6.48}$$

The obtained noise filter above is not square, which is problematic in terms of identification. A spectral factorization of this noise model can be made in order to obtain a square noise filter $H^{(4)}$ and white noise $e^{(4)}$. It is likely that the obtained noise model is then no longer diagonal. The zero-structure of the obtained noise filter can be used as the zero-structure when parameterizing the network model set. Under particular conditions special noise structures can be obtained that can be exploited. If no particular structure is obtained for the noise model, then all process noises are correlated and the applied identification method has to deal with this.

It is now clear how to select measured nodes and external variables such that an abstracted network is obtained with the module of interest invariant. Now a network model set that matches the zero-structure of $G^{(4)}$ and $R^{(4)}$ can be chosen to model the abstracted network. In order to identify the module of interest, for example the joint-direct method can be applied to the full abstracted network such that a maximum likelihood estimate is obtained, or an indirect identification method may be applied. When the objective is to estimate a single module, then applying the joint-direct method to an abstracted network may to be easier to solve than applying the joint-direct method to the original network. However applying the joint-direct method to the abstracted network may still be an overly complicated way of obtaining an estimate of a single module. In future research it is to be investigated how to select nodes such that the simplest estimation problem can be obtained.

6.6 Conclusions

The question to be answered is which nodes are to be measured in order to obtain a consistent estimate of a particular module in a network. As a way to answer this question the concept of abstraction has been introduced as a way to remove unmeasured nodes from a network representation, as a generalization of methods present in literature. In the abstracted network the module of interest remains invariant under some conditions on the chosen measured nodes. Under some additional conditions on the original network structure, e.g. when there is an external excitation present on every node, a network identifiable model set can be defined for the abstracted network in order to identify the module. The results can be applied in combination with different identification methods. In the abstracted network obtained by the abstraction algorithm a transformed network topology is obtained, and from this structure it is clear which nodes and external signals to use when identifying the module. Moreover a systematic method to select which nodes to measure has been introduced.

A-priori assumptions on H and R have been avoided for the formulation of the abstraction method. For the identifiability analysis of the resulting abstracted network some assumptions on R and the network topology have been made. However, this identifiability analysis has been performed with the conservative Theorem 3.10. It is clear that the identifiaility analysis can be extended in future research. With an extension of the identifiability analysis relaxed conditions on the presence of external variables may be obtained. Moreover the identifiability conditions may be formulated on the basis of the network topology. When the identifiability conditions have been relaxed, then a method will be obtained that tells us which nodes need to be measured, and which nodes need to be excited, in order to be able to identify a module of interest.

6.7 Appendix

6.7.1 Proof of Proposition 6.5

Substituting $P = (I - G^{(2)})(I - G^{(1)})^{-1}$ into the definition of the transformation (6.9) gives

$$G^{(2)} = I - (I - G^{(2)})(I - G^{(1)})^{-1}(I - G^{(1)}), \tag{6.49}$$

which shows that $G^{(2)}$ is obtained by applying this transformation. The transfer function matrices $(I - G^{(1)})^{-1}$ and $(I - G^{(2)})$ are both stable and proper. Moreover the diagonal of $(I - P(I - G^{(1)}))$ is 0, so $P = (I - G^{(2)})(I - G^{(1)})^{-1}$ is an appropriate transformation. ■

6.7.2 Proof of Proposition 6.15

In order to prove the proposition the conditions must be interpreted in terms of G. Condition *(a)* and *(b)* imply that there are no direct paths from w_i and w_j to indirect observations $w_{\mathcal{L}}$, i.e.

(i) $G_{\mathcal{L}i} = 0$,

(ii) $G_{\mathcal{L}j} = 0$.

The conditions also imply that there are no paths from w_i and w_j to indirect observations $w_{\mathcal{L}}$ and j that only go through unmeasured nodes $w_{\tilde{\mathcal{Z}}}$, i.e.

(iii) $G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}i} = 0$,

(iv) $G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}j} = 0$,

(v) $G_{j\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}i} = 0$,

(vi) $G_{j\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}j} = 0$.

The module G_{ji} is part of matrix $G_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}}$ which does not get modified in steps 1-2 of Algorithm 6.11, i.e. $G^{(3)}_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}} = G_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}}$. In step 3 the module $G^{(4)}_{ji}$ is defined as

$$G^{(4)}_{ji} = D_{jj}\left(G^{(3)}_{ji} + G^{(3)}_{j\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}i}\right), \tag{6.50}$$

where $D_{jj} = \frac{1}{1-G^{(3)}_{j\mathcal{Z}}(I-G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}j}}$. Now the next steps are to show that

$$0 = G^{(3)}_{j\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}i} \quad \text{and} \tag{6.51}$$

$$0 = G^{(3)}_{j\mathcal{Z}}(I - G^{(3)}_{\mathcal{Z}\mathcal{Z}})^{-1}G^{(3)}_{\mathcal{Z}j}. \tag{6.52}$$

The modules of $G^{(3)}$ are determined in steps 1-2 of Algorithm 6.11 and the relevant modules to show that (6.51) and (6.52) hold are determined next. Modules on row j are left invariant in step 2, such that

$$G^{(3)}_{ji} = G_{ji} \text{ and } G^{(3)}_{j\mathcal{Z}} = G_{j\mathcal{Z}}. \tag{6.53}$$

For the unmeasured nodes there is the factorization

$$G^{(3)}_{\mathcal{Z}\mathcal{Z}} = \begin{bmatrix} G^{(3)}_{\mathcal{V}\mathcal{V}} & G^{(3)}_{\mathcal{V}\tilde{\mathcal{Z}}} \\ G^{(3)}_{\tilde{\mathcal{Z}}\mathcal{V}} & G^{(3)}_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix}, \tag{6.54}$$

where $G^{(3)}_{\mathcal{V}\mathcal{V}} = 0$ in step 2, $G^{(3)}_{\mathcal{V}\tilde{\mathcal{Z}}} = -G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}\tilde{\mathcal{Z}}}$ in step 2, and the modules on rows $\tilde{\mathcal{Z}}$ are left invariant in step 2 $\begin{bmatrix} G^{(3)}_{\tilde{\mathcal{Z}}\mathcal{V}} & G^{(3)}_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix} = \begin{bmatrix} G_{\tilde{\mathcal{Z}}\mathcal{V}} & G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix}$. Then finally there is the factorization

$$G^{(3)}_{\mathcal{Z}i} = \begin{bmatrix} G^{(3)}_{\mathcal{V}i} \\ G^{(3)}_{\tilde{\mathcal{Z}}i} \end{bmatrix}, \text{ and } G^{(3)}_{\mathcal{Z}j} = \begin{bmatrix} G^{(3)}_{\mathcal{V}j} \\ G^{(3)}_{\tilde{\mathcal{Z}}j} \end{bmatrix}, \tag{6.55}$$

where $G^{(3)}_{\mathcal{V}i} = -G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}i}$ and $G^{(3)}_{\mathcal{V}j} = -G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}j}$ in step 2, and the modules on rows $\tilde{\mathcal{Z}}$ are left invariant in step 2 $G^{(3)}_{\tilde{\mathcal{Z}}i} = G_{\tilde{\mathcal{Z}}i}$ and $G^{(3)}_{\tilde{\mathcal{Z}}j} = G_{\tilde{\mathcal{Z}}j}$. Then when rewriting (6.51) and (6.52) using (6.53), (6.54) and (6.55) it must be shown that

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} I & G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}\tilde{\mathcal{Z}}} \\ G_{\tilde{\mathcal{Z}}\mathcal{V}} & G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix}^{-1} \begin{bmatrix} -G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}i} \\ G_{\tilde{\mathcal{Z}}i} \end{bmatrix} \quad \text{and} \tag{6.56}$$

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} I & G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}\tilde{\mathcal{Z}}} \\ G_{\tilde{\mathcal{Z}}\mathcal{V}} & G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix}^{-1} \begin{bmatrix} -G^{-1}_{\mathcal{L}\mathcal{V}}G_{\mathcal{L}j} \\ G_{\tilde{\mathcal{Z}}j} \end{bmatrix}. \tag{6.57}$$

Now using *(i)* and *(ii)* and using the matrix inversion lemma these two equations are rewritten as

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} \star & X \\ \star & Y \end{bmatrix} \begin{bmatrix} 0 \\ G_{\tilde{\mathcal{Z}}i} \end{bmatrix} \quad \text{and} \tag{6.58}$$

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} \star & X \\ \star & Y \end{bmatrix} \begin{bmatrix} 0 \\ G_{\tilde{\mathcal{Z}}j} \end{bmatrix}, \tag{6.59}$$

where

$$X = -\left(I + G_{\mathcal{L}\mathcal{V}}^{-1} G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{V}}\right)^{-1} G_{\mathcal{L}\mathcal{V}}^{-1} G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}$$

and

$$Y = (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} + (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{V}} X.$$

Now using *(iii)* and *(iv)* it can be established that $XG_{\tilde{\mathcal{Z}}i} = 0$ and $XG_{\tilde{\mathcal{Z}}j} = 0$ such that it must be shown that

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} 0 \\ (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}i} \end{bmatrix} \quad \text{and} \tag{6.60}$$

$$0 = \begin{bmatrix} G_{j\mathcal{V}} & G_{j\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} 0 \\ (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}j} \end{bmatrix}, \tag{6.61}$$

which holds by *(v)* and *(vi)*, leaving $G_{ji}^{(4)} = G_{ji}$. ■

6.7.3 Proof of Proposition 6.12

Transfer function matrix $R^{(4)}$ is determined from $u_{\mathcal{S}}^{(4)}$ as

$$R^{(4)} = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D\left(G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} G_{\tilde{\mathcal{Z}}\mathcal{V}}\right) & * \end{bmatrix} R. \tag{6.62}$$

Matrix D is diagonal by definition, matrix $G_{\mathcal{L}\mathcal{V}}$ is diagonal by the condition in the proposition, and $G_{\mathcal{L}\tilde{\mathcal{Z}}} = 0$ by the condition in the proposition, such that the matrix

$$\begin{bmatrix} D & * & 0 & * \\ 0 & * & DG_{\mathcal{L}\mathcal{V}} & * \end{bmatrix} \tag{6.63}$$

can be given a leading diagonal by column operations. Since R is diagonal the same column operations can also be applied to $R^{(4)}$ to obtain a leading diagonal. ■

7 Sequential Least Squares identification algorithm[1]

7.1 Introduction

Algorithms to efficiently solve the identification problem are important for practical applications, therefore the following question is addressed.

Which algorithms are suitable for the efficient identification of large-scale dynamic networks?

In the previous chapters and in literature theory has been developed for the identification of dynamic networks, and in practice this theory used in the form of an algorithm. The algorithms that are deemed suitable must be efficient, where efficient has a double meaning. Firstly efficient refers to statistical efficiency, as in leading to a minimum variance estimate. Secondly efficient refers to an algorithm of low complexity, such that a model can be computed quickly and accurately. Algorithms should be applicable to the small scale problems, such as identification of a local module, but also to the identification of a full large-scale network.

In Chapter 4 a prediction error method to identify dynamic networks is introduced, however the associated cost function (4.8) is typically non-convex. That is, when commonly used model structures as Output Error (OE), AutoRegressive Moving Average with eXogenous input (ARMAX) or Box-Jenkins (BJ) are extended to the dynamic network situation, then this leads to a non-convex cost function prone to local minima. When the size of the network and the number of modules to be estimated grow,

[1] This chapter is based on collaboration with M. Galrinho, G. Bottegal and H. Hjalmarsson, which resulted in the paper (Weerts et al., 2018a).

so does the number of local minima. Moreover, computing the direction of the next iterative solution becomes increasingly challenging. This is particularly important for large-scale dynamic networks, where the large number of parameters may lead to high computational complexity and multiple local minima of the cost function. Algorithms that do not rely on non-convex optimization seem like good candidates for identification of large-scale networks.

In a situation where noise model H is diagonal, the network identification problem can be split into a smaller MISO problem for each node, such as done in Section 2.3. Typical algorithms such as the `ARMAX()` function in Matlab based on (Ljung, 1999), but also subspace algorithms such as SSARX (Jansson, 2003) can be applied to the MISO identification problem. However, when process noises are correlated, and there are loops in the network, then estimation of MISO models with a direct method leads to biased estimates. In such a situation the joint-direct method from Chapter 4 can be applied, which requires solving a MIMO problem. In this MIMO setting, nodes that appear as an input do not have to influence every node that appears as ouput. There is a need to encode the network topology specified by the network model set in the algorithm. It is not possible to encode the network topology in the SSARX algorithm in the MIMO situation, and therefore this algorithm is not suitable for network identification. Subspace identification methods that are able to encode the network topology exist, e.g. (Torres et al., 2015), however these operate under restrictive assumptions on the process noise and are therefore not considered.

One of the difficulties of the correlated noise is that the $H(q,\theta)$ becomes non-diagonal. In the prediction error (4.7) the inverse $H^{-1}(q,\theta)$ appears, which can cause difficulties. When H has some particular structure, this structure is typically lost in the inverted H^{-1}, possibly resulting in H^{-1} not having any 0 elements. Moreover the parameterization of H^{-1} will be fairly complicated.

Instrumental variables methods (Dankers et al., 2015) or two-stage methods (Van den Hof et al., 2013) do not need an accurate noise model in order to obtain consistent estimates. This implies that a non-diagonal H does not have to be modeled, and a MISO setting can be used regardless of noise correlation. These methods can make use of linear-in-the-parameters models, and can thus lead to convex optimization problems. However external excitations that are sufficiently informative and of sufficient power are a necessary requirement for these methods to succeed. An additional issue is that these methods use the external excitation to de-correlate noise contributions, such that excitation provided by process noise is not used to reduce variance, resulting in estimates that are not minimum variance. It can be concluded that the classical methods and algorithms are not sufficient for minimum variance identification of large-scale dynamic networks.

In (Everitt, 2017) Model Order Reduction Steiglitz-McBride (MORSM) is introduced as a method to estimate modules in a network. The method is similar to a two-stage identification approach (Van den Hof et al., 2013) and leads to an algorithm based on the solving of a sequence of least squares problems. Since only least squares problems are solved, the method relies on convex optimizations and does not run into local minima. However since the method resembles a two-stage approach, it suffers from increased variance. Additionally the algorithm needs to iterate an infinite number of times in order to achieve consistency. The method did provide the inspiration for the approach developed in this chapter.

For single-input single-output (SISO) ARMA time-series, (Durbin, 1960) observed that if the innovations sequence is known, the model can be written as a linear regression model, and the model parameters estimated with a least-squares criterion. Based on this, he proposed a method—Durbin's first method—where the innovations are first estimated as the residuals of a high-order AR-model, which are then used to estimate the ARMA parameters using another least-squares criterion. This method is not asymptotically efficient, which was remedied in (Mayne and Firoozan, 1982) by filtering the output and the AR residuals with the inverse of the estimated MA-polynomial obtained from Durbin's first method, and then re-estimating the ARMA-parameters. The asymptotic results in Mayne and Firoozan (1982) are not entirely satisfactory, since they do not cover the situation where the number of parameters in the AR model is a function of the number of available data samples, which is addressed in Hannan and Kavalieris (1983). This type of method has become popular for vector ARMA time series (e.g., Dufour and Jouini (2014) and references therein) due to their computational and optimal statistical properties.

Durbin's method will be extended for the identification of ARMAX dynamic network models. The flexibility of the method allows to encode the network topology and capture the noise correlations. A close relation between the introduced method and the method of Weighted Null Space Fitting (WNSF) of (Galrinho et al., 2014) will be shown. As a result the thorough asymptotic analysis in (Galrinho et al., 2017) can be used to show the properties of the proposed method.

The chapter proceeds with a definition of the parametric network ARMAX model and the identification algorithm in Section 7.2. Asymptotic properties of this algorithm are investigated in Section 7.3. Issues regarding practical implementation and simulations showing the performance of the algorithm are presented in Section 7.4. Finally possible extensions are discussed and conclusions are drawn.

7.2 Sequential Least Squares algorithm

7.2.1 Parametric ARMAX network model

The network model set $\mathcal{M}$ defined in Definition 2.7 is assigned a specific parameterization. The parameterized rational transfer functions relate to matrix polynomials

$$\left\{\begin{matrix} G(q,\theta) = D^{-1}(q,\theta)N_G(q,\theta), \\ H(q,\theta) = D^{-1}(q,\theta)N_H(q,\theta), \\ R(q,\theta) = D^{-1}(q,\theta)N_R(q,\theta), \end{matrix}\right\} \tag{7.1}$$

where D, N_G, N_H are proper polynomial matrices in q^{-1} of order n_p. In order for the algorithm to work some restrictions must be introduced to the parameterization:

- $D(q,\theta)$ is diagonal and monic,
- $N_G(q,\theta)$ has zeros on the diagonal,
- $N_H(q,\theta)$ is monic.

It is assumed that the network model set is globally network identifiable at $M(q,\theta_0)$.

As D is diagonal, all transfer functions in one row of $\begin{bmatrix} G & R & H \end{bmatrix}$ have the same poles. With adequate model order n_p this model structure is able to capture any data generating network represented in (2.4). Hence it is not restrictive to assume that there exists a parameter θ_0 for which the model captures all dynamics in the network, i.e. $\mathcal{S} = M(\theta_0)$.

It should be noted that the network ARMAX model is related to typical open-loop MIMO ARMAX models of the form

$$\mathcal{A}(q,\theta)w(t) = \mathcal{B}(q,\theta)r(t) + \mathcal{C}(q,\theta)e(t). \tag{7.2}$$

The network and open-loop models can be related through

$$D(q,\theta)w(t) = N_G(q,\theta)w(t) + N_R(q,\theta)r(t) + N_H(q,\theta)e(t), \tag{7.3}$$

such that

$$\mathcal{A} = D - N_G, \quad \mathcal{B} = N_R, \quad \mathcal{C} = N_H. \tag{7.4}$$

Since $N_G(q,\theta)$ has zeros on the diagonal and $D(q,\theta)$ is diagonal the parameters do not mix in $\mathcal{A}$. The network ARMAX and open-loop ARMAX models are typically used in different ways, as the open-loop model is typically unstructured in contrast to the network model that contains the network topology as a structure.

Due to the parameterization defined in (7.1) and the diagonal structure of D^{-1}, the zero-structure of $G(q,\theta)$ is the same as the zero-structure of $N_G(q,\theta)$, i.e.

$$G_{ji}(q,\theta) = 0 \Leftrightarrow (N_G)_{ji}(q,\theta) = 0. \tag{7.5}$$

The same principle holds for the zero-structures of R, H and N_R, N_H. This implies that when the zero-structure can be encoded in a standard ARMAX algorithm, then this algorithm can be used to estimate network ARMAX models.

7.2.2 Step 1: ARX fitting of network dynamics

This step serves to make an initial estimate of the dynamics, without taking the network structure into account. An ARX model structure is defined as

$$\mathcal{M}_A = \left\{ M_A = (A(q,\eta), B(q,\eta)), \eta \in \beta \right\},$$

where A and B are $L \times L$ proper polynomial matrices of order n_A, and $A(q,\eta)$ is monic. This model is a parameterization of the open-loop response in the following way

$$T_{wr}(q,\eta) = A^{-1}(q,\eta)B(q,\eta), \quad T_{we}(q,\eta) = A^{-1}(q,\eta). \tag{7.6}$$

Using this ARX model a prediction error is defined as

$$\varepsilon_A(t,\eta) = A(q,\eta)w(t) - B(q,\eta)r(t), \tag{7.7}$$

and optimized with the weighted least squares criterion

$$\hat{\eta}_N = \arg\min_{\eta\in\beta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_A^T(t,\eta)\; Q_A\; \varepsilon_A(t,\eta)\;, \quad Q_A > 0. \tag{7.8}$$

This estimate has an analytical closed-form solution.

When the ARX estimate is consistent, we also have reconstructed the innovation of the network as $\hat{\varepsilon}_A(t) := \varepsilon_A(t,\hat{\eta}_N)$. The error between the innovation and the reconstructed innovation is denoted by

$$s(t) := e(t) - \hat{\varepsilon}_A(t) \tag{7.9}$$

and will be used in the later steps of the algorithm.

7.2.3 Step 2: Reconstructed innovation as input

From (7.9), we notice that the innovation $e(t)$ can be written as a sum of a known signal $\hat{\varepsilon}_A(t)$ and an unknown signal $s(t)$. Substituting this into (2.4) yields

$$w = G^0 w + R^0 r + H^0 \hat{\varepsilon}_A + H^0 s. \tag{7.10}$$

Since $\hat{\varepsilon}_A(t)$ is known, it acts as an input in the above network formulation, while s acts as the "new innovation". Note that $s(t)$ becomes smaller when the innovation is estimated better. If the innovation is recovered exactly, $s(t) = 0$ for all time and the above network essentially is noise-free and deterministic. The related prediction error is

$$\varepsilon_s(t,\theta) = H^{-1}(q,\theta)\big((I - G(q,\theta))w(t) - R(q,\theta)r(t)\big) - \varepsilon_A(t,\hat{\eta}_N). \tag{7.11}$$

When observing from (7.1) that $G(q,\theta)$, $H(q,\theta)$, $R(q,\theta)$ share a common denominator $D(q,\theta)$, the related prediction error is

$$\varepsilon_s(t,\theta) = N_H^{-1}(q,\theta)\varepsilon_{L2}(t,\theta), \tag{7.12}$$

with

$$\varepsilon_{L2}(t,\theta) \;=\; (D(q,\theta) \;-\; N_G(q,\theta))w(t) \;-\; N_R(q,\theta)r(t) \;-\; N_H(q,\theta)\varepsilon_A(t,\hat{\eta}_N). \tag{7.13}$$

Note that D is diagonal and N_G is 0 on the diagonal such that their parameterizations do not mix. The estimated innovation $\hat{\varepsilon}_A$ acts as an additional input, parameterized with the same parameters as the noise model. The N_H has the same structure as H, and so structure restrictions from H also need to be encoded into N_H. Polynomial matrix N_H is parameterized, and not its inverse, this means that structure restrictions on N_H can easily be parameterized in this estimation algorithm.

The relation between the original prediction error defined in (4.7) and new prediction error is

$$\varepsilon_s(t,\theta) = \varepsilon(t,\theta) - \hat{\varepsilon}_A, \tag{7.14}$$

so ε_s is non-linear in the parameters just like ε. However, we note that ε_{L2} is linear in the parameters. Instead of optimizing over ε_s, estimation is defined as

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_{L2}^T(t,\theta)\; Q\; \varepsilon_{L2}(t,\theta) \quad , Q > 0, \tag{7.15}$$

which has a closed-form solution in θ. An estimate of all the ARMAX polynomials is obtained, including the numerator of the noise model N_H. For SISO ARMA models, (7.15) corresponds to Durbin's first method (Durbin, 1960).

7.2.4 Step 3: Improve approximation

In step 2, an approximation of $\varepsilon_s(t,\theta)$ has been made in order to obtain a convex criterion that yields an estimate of the parameters. Using the estimate of N_H from step 2, we can construct a new criterion to refine the parameter estimates. To do so, we define a new approximation of $\varepsilon_s(t,\theta)$, where the parameterized term $N_H^{-1}(q,\theta)$ is replaced with the estimated version from the previous step. This can be done for one step, or optionally in an iterative procedure as follows: For $k \geq 3$ use:

$$\varepsilon_{Lk}(t,\theta) := N_H^{-1}(q,\hat{\theta}_N^{[k-1]})\varepsilon_{L2}(t,\theta). \tag{7.16}$$

Then we can define criterion

$$\hat{\theta}_N^{[k]} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_{Lk}^T(t,\theta)\; Q\; \varepsilon_{Lk}(t,\theta). \tag{7.17}$$

The algorithm can be summarized as follows.

> **Algorithm 7.1.**
>
> 1) *Choose an ARX model set $\mathcal{M}_A$ with model order n_A, and a parametric network model set $\mathcal{M}$ with model order n.*
>
> 2) *Solve the multivariable linear regression problem (7.8) with w as the output and r as the input, while using model set $\mathcal{M}_A$.*
>
> 3) *Compute $\hat{\varepsilon}_A = A(\hat{\eta}_N)w - B(\hat{\eta}_N)r$.*
>
> 4) *Solve the linear regression problem (7.15) where w are the nodes and r and $\hat{\varepsilon}_A$ are the inputs, while using network model set $\mathcal{M}$.*
>
> 5) *Set $k = 3$*
>
> 6) *Solve the linear regression problem (7.17) where the error of step 4) is pre-filtered with $N_H^{-1}(q,\hat{\theta}_N^{k-1})$, while using network model set $\mathcal{M}$.*
>
> 7) *If no stopping criterion is reached, increase k by 1, and return to 6).*

Some different stopping criteria maybe used for the algorithm above, for example when $\|\hat{\theta}_N^{[k]} - \hat{\theta}_N^{[k-1]}\|_2$ is below a threshold value, or when a fixed number of iterations is reached.

7.2.5 Discussion

In most practical situations the ARX estimate can only be consistent when the model order $n_A \to \infty$. With finite data length the number of parameters can not be infinite due to a lack of data. An approximation of the data generating system has to be made by choosing an ARX model of high, but not infinite, order, which causes the estimated innovation to be an approximation. The consequences of a finite ARX model order are investigated in a later section.

The estimated innovation $\hat{\varepsilon}_A$ is used as an input in every subsequent identification step. Any difference between estimated and true innovation propagates through each iteration of the algorithm. This has the undesired effect that the quality of the ARX estimate limits the quality of the final parametric estimate.

In Algorithm 7.1 the ARX model is estimated first, and the next steps depend on the estimated ARX model. But we are not directly interested in a high quality ARX model, it only serves as an intermediate step to obtain a parametric ARMAX model. A possible topic for further research is to perform a joint optimization of the ARX and ARMAX models. The reason that joint optimization is interesting is that the parametric restrictions imposed by the ARMAX model will reduce the variance of the ARX model, which possibly improves the estimated model. After obtaining initial ARX and ARMAX estimates, this joint optimization is to jointly optimize the functions

$$V_k(\theta,\eta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_{Lk}^T(t,\theta,\eta) Q \varepsilon_{Lk}(t,\theta,\eta), \text{ and } V_A(\eta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_A^T(t,\eta) Q_A \varepsilon_A(t,\eta) \tag{7.18}$$

with appropriate weighting. Setting the weight on the ARX objective function V_A to infinity would be equivalent to Algorithm 7.1. Note that $\varepsilon_{Lk}(t,\theta,\eta)$ in this joint optimization has bi-linear parameterization since the $\varepsilon_A(t,\eta)$ would now be parameterized, causing multiplication of the η and θ parameters. An initial guess for the obtained non-convex criterion is available from the Sequential Least Squares algorithm as defined before.

7.3 Asymptotic properties

7.3.1 Sketch of consistency

Although informal, the following argument provides an intuition for consistency of Algorithm 7.1. A reasoning is provided under the assumption that a consistent ARX estimate is obtained. In that situation the ARX prediction error converges to the innovation, i.e. $\varepsilon_A(t,\hat{\eta}_N) \to e(t)$ with probability 1 for $N \to \infty$. The estimated innovation sequence is used as a known input in the ARMAX estimation step (7.15), and the $s(t)$ acts as the "new innovation". We can investigate whether θ_0 is a minimum of (7.15). When substituting the true network into ε_{L2}, we obtain the expression

$$\begin{aligned}\varepsilon_{L2}(t,\theta) =&\Big(X(\theta)N_R(\theta_0) - N_R(\theta)\Big)r(t)\\ &+ \Big(X(\theta)N_H(\theta_0) - N_H(\theta)\Big)\hat{\varepsilon}_A(t)\\ &+ \ X(\theta)N_H(\theta_0)s(t),\end{aligned} \tag{7.19}$$

driven only by external signals, with

$$X(\theta) = (D(\theta) - N_G(\theta))(D(\theta_0) - N_G(\theta_0))^{-1}. \tag{7.20}$$

We can see that $X = 1$ when $\theta = \theta_0$, and then the first two terms of (7.19) are 0, and so $\varepsilon_{L2}(t,\theta_0) = N_H(\theta_0)s(t)$. Due to consistency of the ARX estimate,

$$s(t) = e(t) - \varepsilon_A(t,\hat{\eta}_N) \to 0, \tag{7.21}$$

which implies that $\varepsilon_{L2}(t,\theta_0) \to 0$. Then, the cost function of (7.15) is 0 and minimized by the true network θ_0. When θ_0 is also the only minimum of the cost function, consistency can be proven.

7.3.2 Connection to WNSF

A more thorough way of showing consistency of Algorithm 7.1 is developed here by relating Algorithm 7.1 to the WNSF algorithm (Galrinho et al., 2014). To this end first the WNSF algorithm is defined, after which it is related to Algorithm 7.1 such that then the asymptotic properties can be obtained.

The WNSF algorithm has been defined as an estimation algorithm for SISO systems (Galrinho et al., 2014, 2017). Different parametric model structures can be handled by WNSF, e.g. Output Error, Box-Jenkins and ARMAX. Here the ARMAX version of WNSF will be formulated for the situation that there is one node w and one external excitation, i.e. $w \in \mathbb{R}$, $r \in \mathbb{R}$. However the algorithm can also be applied for a situation where one node is the output, and another node is the input, simply by labeling the input as r.

As a first step an ARX model of order n_A is estimated exactly as done in Section 7.2.2. The obtained ARX model is characterized by

$$A(\eta_A) \in \mathbb{R}(q^{-1}), \quad B(\eta_B) \in \mathbb{R}(q^{-1}), \tag{7.22}$$

where $\eta_A \in \mathbb{R}^{n_A}$, $\eta_B \in \mathbb{R}^{n_A+1}$. With sufficiently high order n_A, the ARX model captures the open-loop response of the network, i.e.

$$T_{wr}(q) = \frac{B(q,\eta_B)}{A(q,\eta_A)}, \quad T_{we}(q) = \frac{1}{A(q,\eta_A)}, \tag{7.23}$$

for some parameters η_A and η_B.

An ARMAX network model is characterized as in (7.1), and this relates to an ARMAX model as described in (7.4). With sufficiently high order n_p, the ARMAX model captures the open-loop response of the network, i.e.

$$T_{wr}(q) = \frac{\mathcal{B}(q,\theta_B)}{\mathcal{A}(q,\theta_A)}, \quad T_{we}(q) = \frac{\mathcal{C}(q,\theta_C)}{\mathcal{A}(q,\theta_A)}, \tag{7.24}$$

for some parameters θ_A, θ_B and θ_C. It would be possible to write the above equation in terms of $D(\theta), N_G(\theta), N_R(\theta), N_H(\theta)$, but this would make the notation in the remainder of the section even more involved than it is. The WNSF is based on the relation between the ARX and ARMAX models that follows from (7.23) and (7.24)

$$\frac{B(q,\eta_B)}{A(q,\eta_A)} = \frac{\mathcal{B}(q,\theta_B)}{\mathcal{A}(q,\theta_A)}, \tag{7.25}$$

$$\frac{1}{A(q,\eta_A)} = \frac{\mathcal{C}(q,\theta_C)}{\mathcal{A}(q,\theta_A)}. \tag{7.26}$$

By filling (7.26) in into (7.25) it is directly obtained that

$$\mathcal{C}(q,\theta_C)B(q,\eta_B) - \mathcal{B}(q,\theta_B) = 0. \tag{7.27}$$

By multiplying with both denominators (7.26) is transformed into

$$\mathcal{C}(q,\theta_C)A(q,\eta_A) - \mathcal{A}(q,\theta_A) = 0. \tag{7.28}$$

When the parameters η_A and η_B are given, or estimated beforehand, then (7.27) and (7.28) are linear in the parameters $\theta_A, \theta_B, \theta_C$. For each polynomial order in q^{-1} an equation can be obtained that relates linearly to the parameters in θ_A, θ_B and θ_C, which is illustrated by writing the two equations in regression form. The polynomials consist of terms that are multiplied with $q^0, \cdots, q^{n_A+n_p}$, and an equation will be associated with each of these terms. The following notation will be used: Let $\mathcal{T}(x)$ be a lower-triangular Toeplitz matrix whose first column is x. Then (7.27) is represented by

$$\mathcal{T}\left(\begin{bmatrix}\eta_B\\ 0_{n_p\times 1}\end{bmatrix}\right)\begin{bmatrix}1\\ \theta_C\end{bmatrix} - \begin{bmatrix}\theta_B\\ 0_{(n_A)\times 1}\end{bmatrix} = 0, \tag{7.29}$$

where $0_{(n_A\times 1)}$ denotes a column vector of zero of dimension $n_A \times 1$, and where the Toeplitz matrix is of dimension $(1+n_A+n_p)\times(n_p+1)$, and (7.28) is represented by

$$\mathcal{T}\left(\begin{bmatrix}1\\ \eta_A\\ 0_{n_p\times 1}\end{bmatrix}\right)\begin{bmatrix}1\\ \theta_C\end{bmatrix} - \begin{bmatrix}1\\ \theta_A\\ 0_{(n_A)\times 1}\end{bmatrix} = 0, \tag{7.30}$$

with the Toeplitz matrix also of dimension $(1+n_A+n_p)\times(n_p+1)$. Now (7.29) and (7.30) are written in regression form as

$$\begin{bmatrix}1\\ \eta_A\\ 0_{n_p\times 1}\\ \eta_B\\ 0_{n_p\times 1}\end{bmatrix} + \underbrace{\begin{bmatrix}-\mathcal{T}\left(\begin{bmatrix}1\\ 0_{(n_A+n_p)\times 1}\end{bmatrix}\right) & 0 & \mathcal{T}\left(\begin{bmatrix}1\\ \eta_A\\ 0_{n_p\times 1}\end{bmatrix}\right)\\ 0 & -\mathcal{T}\left(\begin{bmatrix}1\\ 0_{(n_A+n_p)\times 1}\end{bmatrix}\right) & \mathcal{T}\left(\begin{bmatrix}\eta_B\\ 0_{n_p\times 1}\end{bmatrix}\right)\end{bmatrix}}_{:=Q(\eta)}\begin{bmatrix}1\\ \theta_A\\ \theta_B\\ \theta_C\end{bmatrix} = 0, \tag{7.31}$$

where the term $\begin{bmatrix}1 & \eta_A^T & 0_{n_p\times 1}^T & \eta_B^T & 0_{n_p\times 1}^T\end{bmatrix}^T$ is of dimension $2(n_A+n_p+1)\times 1$. Let the residual ε_η of dimension $2(n_A+n_p+1)\times 1$ denote

$$\varepsilon_\eta(\eta,\theta) := \begin{bmatrix}1\\ \eta_A\\ 0_{n_p\times 1}\\ \eta_B\\ 0_{n_p\times 1}\end{bmatrix} + Q(\eta)\begin{bmatrix}1\\ \theta_A\\ \theta_B\\ \theta_C\end{bmatrix}. \tag{7.32}$$

Then the regression form is related to (7.27) and (7.28) via

$$\varepsilon_\eta^T(\eta,\theta)\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix} = \begin{bmatrix}\mathcal{C}(q,\theta_C)A(q,\eta_A)-\mathcal{A}(q,\theta_A) & \mathcal{C}(q,\theta_C)B(q,\eta_B)-\mathcal{B}(q,\theta_B)\end{bmatrix}, \tag{7.33}$$

where $\rho = \begin{bmatrix}1 & q^{-1} & \cdots & q^{-n_A-n_p}\end{bmatrix}^T$.

The WNSF estimation is then defined as an optimization over θ of the residual ε_η with the estimated $\hat{\eta}$ as an argument

$$\hat{\theta}_{WNSF}^{[k]} = \arg\min_{\theta\in\Theta} \varepsilon_\eta^T(\hat{\eta}_N,\theta)W(\hat{\theta}_N^{[k-1]})\varepsilon_\eta(\hat{\eta}_N,\theta), \tag{7.34}$$

where $W(\hat{\theta}_{WNSF}^{[k-1]})$ is an appropriate weight that depends on the previous estimate $\hat{\theta}_N^{[k-1]}$ or an initialization. The weight $W(\theta)$ is defined as follows

$$W(\theta) := T_C^{-1}(\theta)PT_C^{-T}(\theta), \tag{7.35}$$

where

$$T_C(\theta) := \begin{bmatrix}\mathcal{T}\left(\begin{bmatrix}1\\ \theta_C\\ 0_{n_A\times 1}\end{bmatrix}\right) & 0\\ 0 & \mathcal{T}\left(\begin{bmatrix}1\\ \theta_C\\ 0_{n_A\times 1}\end{bmatrix}\right)\end{bmatrix}, \tag{7.36}$$

with the Toeplitz matrices of dimension $(1+n_A+n_p)\times(1+n_A+n_p)$, and where

$$P = \frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}^T\begin{bmatrix}w(t)\\ r(t)\end{bmatrix}\begin{bmatrix}w^T(t) & r^T(t)\end{bmatrix}\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}. \tag{7.37}$$

The interpretation of P is a weighting with the data spectrum. The interpretation of T_C^{-1} is a weighting with the inverse of $\mathcal{C}$, which is made on the basis of the previous estimate. It should be noted that the toeplitz matrix in T_C^{-1} corresponds to a filter constructed from the first n_A+n_p+1 impulse response coefficients of $\mathcal{C}^{-1}$. This means that there is a small approximation of the filter $\mathcal{C}^{-1}$ due to truncation, but when the order n_A+n_p+1 is large enough this error is insignificant, and it disappears when the ARX order $n_A \to \infty$.

As an initialization an initial weight must be chosen, for example

$$W = I, \tag{7.38}$$

as a pure least square estimator, or

$$W(0) = P \tag{7.39}$$

to weight the data spectrum in the estimate. The WNSF algorithm can then be summarized in the following algorithm.

Algorithm 7.2 (SISO WNSF ARMAX). *Let there be one node w that acts as output and one external variable r that acts as input.*

1) Choose an ARX model set $\mathcal{M}_A$ with model order n_A, and a parametric network model set $\mathcal{M}$ with model order n.

2) Solve the multivariable linear regression problem (7.8) with w as the output and r as the input, while using model set $\mathcal{M}_A$.

3) Set $k = 2$ and compute the initial weight as either (7.38) *or* (7.39)*.*

4) Increase k by 1.

5) Solve the linear regression problem (7.34) *with the computed weight.*

7) If not converged, compute $W(\hat{\theta}^{[k]}_{WNSF})$ from (7.35) *and return to 4).*

Now that the WNSF algorithm is defined, the equivalence relation with the Seqential Least Squares algorithm can be shown.

Proposition 7.3. *Consider estimates obtained by Sequential Least Squares in Algorithm 7.1 and Weighted NullSpace Fitting in Algorithm 7.2 when initialized by* (7.39) *for a "SISO" network with precisely one node w_1 and one external excitation r_1.*

1. The initial estimates are equal, i.e.

$$\hat{\theta}^{[2]}_N = \hat{\theta}^{[2]}_{WNSF}. \tag{7.40}$$

2. The further iterations are approximately equal, i.e.

$$\hat{\theta}^{[k]}_N \approx \hat{\theta}^{[k]}_{WNSF}, \quad k \geq 3, \tag{7.41}$$

due to truncation of the impulse response of $\mathcal{C}^{-1}(\hat{\theta}^{[k-1]}_{WNSF})$ in the WNSF estimate.

Proof. Provided in Appendix 7.6.1. ■

The impact of truncation of the impulse response of $\mathcal{C}^{-1}(\hat{\theta}^{[k-1]}_N)$ is negligible in practice, as the values are nearly 0 at the point of truncation. This difference has been treated in (Galrinho, 2016) for OE models. Moreover when $n_A + n_p + 1 \to \infty$ then the difference disappears.

Equivalence of Algorithm 7.2 and Algorithm 7.1 has some implications. Due to the equality relations derived in Proposition 7.3, for a SISO situation step 2 of SLS is a consistent estimate, and step 3 is an asymptotically efficient estimate. This is unlike other iterative methods, e.g. MORSM (Everitt, 2017), where an infinite number of iterations are required in order to obtain consistency.

Despite the equivalence relations between SLS and WNSF there is added value in the SLS algorithm. SLS serves as an alternative interpretation of the WNSF algorithm, and this interpretation shows how WNSF is connected to the non-convex prediction error criterion. In terms of execution the SLS may not require the explicit computation of $\mathcal{C}^{-1}$ if the errors are implicitly filtered by this inverse, whereas in WNSF the impulse response of $\mathcal{C}^{-1}$ is explicitly used in the weight of the third step. Moreover SLS can be executed by sequentially calling the `ARX()` command in Matlab, although with different model sets and signals in each call.

Given that Algorithm 7.1 is asymptotically the same as WNSF in the SISO situation, consistency and asymptotic efficiency in the MIMO situation follow from extending the results of Proposition 7.3 and the asymptotic analysis in (Galrinho et al., 2017) to MIMO ARMAX models. This extension is left as a future work.

7.4 Implementation issues and simulations

7.4.1 Selection of ARX order

For typical networks the order of the ARX model must go to infinity in order to obtain consistent estimates. In practice a data set is of finite length, and so an infinite model order is not feasible. There will be bias, and there will be variance, so it is necessary to tune the order of the ARX model such that a satisfactory trade-off is achieved between bias and variance.

The error of the ARX model, either in bias or variance, can directly be observed in σ_s^2, the power of $s(t)$. When the number of parameters is 0, then $\hat{\varepsilon}_A(t) = w(t)$ and there is a large bias error in σ_s^2. On the other hand when the number of parameters is N, then $\hat{\varepsilon}_A(t) = 0$ and there is a large variance error σ_s^2. Computing the optimal σ_s^2 is not possible from just estimation data, since it requires knowledge of the true innovation sequence. Therefore a method is required that performs the bias-variance trade-off for us in a smart way.

One way of choosing the ARX order is by checking whether the residuals pass a correlation test, and if not then increase the order. This iterative procedure of model order selection may be time consuming, and depends on user knowledge and experience. In order to prevent manual tuning we will only consider systematic methods to select the ARX order. In particular we consider Akaike's Information Criterion (AIC), and selecting the optimal ARX order on the basis of the PEM criterion. More formally the AIC criterion is the minimization of

$$AIC(n) = \log\left(\frac{1}{N}\sum_{t=1}^{N}\varepsilon_A^2(t,\eta)\right) + \frac{n}{N}, \tag{7.42}$$

where $n = \dim(\eta)$ is the number of parameters in the ARX model. The function $AIC(n)$ can be minimized by repeating step 1 of Algorithm 7.1 for different model orders n_A. Alternatively the algorithm steps 2 and 3 can be applied for every model order n_A, after which the performance of the ARMAX models is evaluated on

$$V(n_A) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^T(t, \hat{\theta}(n_A))\, Q\, \varepsilon(t, \hat{\theta}(n_A)), \tag{7.43}$$

where $\hat{\theta}(n_A)$ is the estimate in Step 3 that belongs to the ARX model of order n_A. The optimal model order n_A is the one that minimizes $V(\hat{\theta}(n_A))$. A difference between the two methods is that with AIC the order of the ARX model can be determined without performing the ARMAX modeling steps, saving computational load. Differences in performance for these two selection methods are illustrated in simulation.

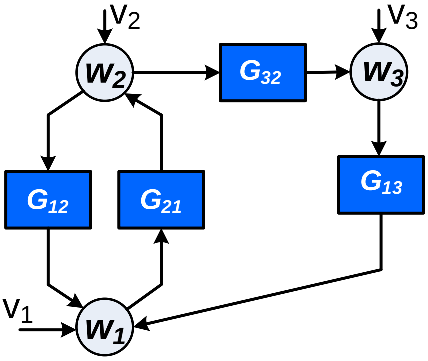


Figure 7.1: A 3 node network used for simulations.

The network used to evaluate the performance of the order selection methods is depicted in Figure 7.1. The data generating network model is described by the model $M^0 = (G^0, H^0, \Lambda^0)$, which has the following dynamics

$$\begin{aligned}
G_{32^0} &= \frac{q^{-1} + 0.5q^{-2}}{1 - 0.5q^{-1} + 0.2q^{-2}}, & H_{33}^0 &= \frac{1 - 0.6q^{-1}}{1 - 0.85q^{-1}},\\
G_{21}^0 &= \frac{0.4q^{-1} - 0.2q^{-2}}{1 + 0.4q^{-1} - 0.5q^{-2}}, & H_{22}^0 &= \frac{1 - 0.3q^{-1}}{1 - 0.9q^{-1}},\\
G_{13}^0 &= \frac{0.8q^{-1}}{1 - 0.3q^{-1}}, & H_{11}^0 &= \frac{1 + 0.5q^{-1}}{1 - 0.7q^{-1}},\\
G_{12}^0 &= \frac{-0.7q^{-1}}{1 - 0.7q^{-1}},
\end{aligned}$$

and $\Lambda = I$. Since H is diagonal, the joint-direct method consists of 3 separate MISO problems that can be solved separately. The measure to evaluate performance of the order selection methods is how well module G_{32}^0 is estimated. Performance is evaluated by testing the fit of the impulse response onto the true impulse response, i.e. using the fit ratio defined by

$$f_g(\theta) = 1 - \frac{\|g(\theta_0) - g(\theta)\|_2}{\|g(\theta_0)\|_2}, \tag{7.44}$$

where g is the impulse response of a module. A 3rd order ARMAX model is estimated for node 3, which can exactly capture the data generating system M^0, $N = 1000$ datapoints are available, and 100 Monte-Carlo runs are performed.

The ARX order of the model is varied from 3 to 50 and the 10th iteration of Algorithm 7.1 is used to determine a model for every ARX order. Using the two criteria (7.42) and (7.43) the best models are selected. In Figure 7.2 the fit onto the true module is compared between the two selection methods and the model estimated by `ARMAX()` initialized in the true network. It can be concluded that selection of the ARX order by the PEM criterion (7.43) is the better selection method, and for this example system no further improvement seems possible for ARX order selection. When selecting the ARX order by the AIC criterion the performance deteriorates slightly. However there may be some value in the AIC selection method due to the lower computational load. When α denotes the number of different orders that are to be selected, and β is the number of iterations to estimate the ARMAX model, then with the order selection based on AIC only α least squares problems have to be solved, while with the PEM criterion $\alpha(1 + \beta)$ least squares problems have to be solved.

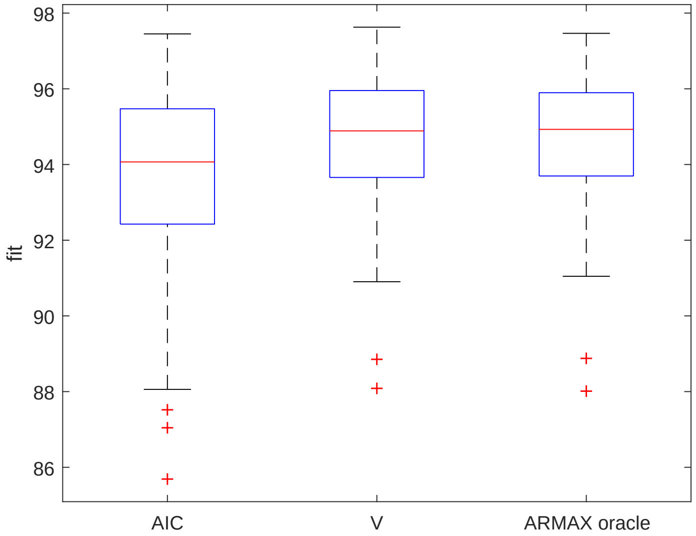


Figure 7.2: Boxplot to compare ARX order selection methods on the basis of the fit of module G^0_{32}. Left) Order selection by AIC criterion, Middle) Order selection by PEM criterion, Right) Benchmark with Matlab's `ARMAX()` algorithm initialized at the data generating system as comparison.

Remark 7.4. *In Algorithms 7.1 and 7.2 there is no guarantee on stability of N_H^{-1}. An unstable N_H^{-1} will lead to bad estimates in step 3 and further iterations. For most selections of ARX order n_A the algorithms result in a stable N_H^{-1}, but for some selections of the ARX order an unstable result is obtained. Models with unstable N_H^{-1} lead to a large value in the PEM criterion, and consequently the ARX orders that lead to unstable models are not chosen as the final estimate.*

7.4.2 Selection of iteration

This section is dedicated to the iterations that Algorithm 7.1 can perform, in order to improve estimates. At $k = 3$ the estimate obtained in *6)* of Algorithm 7.1 is asymptotically efficient, and it is claimed that further iterations can improve the estimate. A simulation is performed to verify this claim.

Simulations using the network model M^0 in the previous section as data generating system are performed. For these simulations the order of the ARX model is fixed at a sufficiently high order $n_A = 35$. Now estimates are made on different data lengths, to evaluate the convergence of the error. For each different data length N, 100 Monte-Carlo runs are performed. As the performance measure, the difference in impulse response between estimate and true module is averaged over the 100 Monte-Carlo runs, i.e.

$$MSE(N) = \frac{1}{100} \sum_{i=1}^{100} \|g(\theta_0) - g(\theta_i(N))\|_2, \tag{7.45}$$

where $\theta_i(N)$ indicates the estimated for the i-th Monte-Carlo run for data length N. Fig. 7.3 shows the resulting average MSE per data length N for the 2nd, 3rd and 20-th iteration of Algorithm 7.1, and for the joint-direct estimate computed by the Matlab algorithm `armax()` using the true system as initialization.

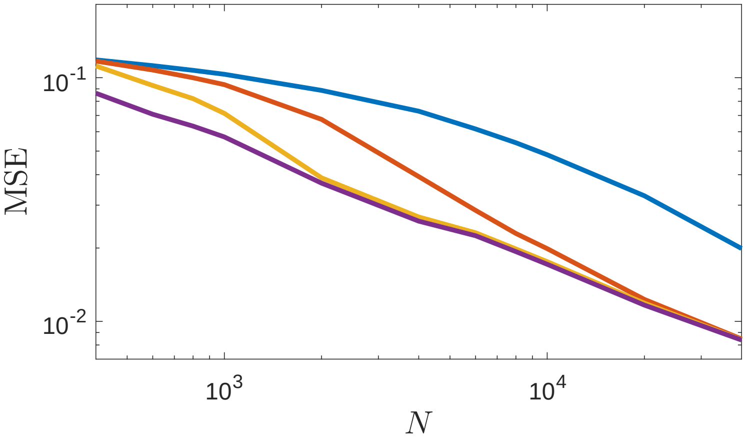


Figure 7.3: MSE over 100 Monte-Carlo runs plotted against data length N. Blue: Step 2, Red: Step 3, Yellow: 20-th iteration, and Purple: the joint direct method.

For small numbers of data N, Algorithm 7.1 has a slightly higher MSE than the joint direct method on average. Increasing N leads to improved models for each step of the algorithm. Around $N = 3 \cdot 10^4$, step 3 has the same MSE as PEM, and around $N = 6 \cdot 10^3$ the 20-th iteration has the same MSE as PEM. On average continued iterations improve the estimate of the algorithm. The point of these simulations is to show that Algorithm 7.1 is a close approximation of the non-convex optimization problem, even for small data sets.

Remark 7.5. *A comparison between the SLS and WNSF algorithms has been made by estimating models with Algorithm 7.2, with initial weight P, on the basis of the same data sets as used in Figure 7.3. The MSE of the models estimated with WNSF for step 2,3 and iteration 10 result in an MSE plot that is indistinguishable from the plot depicted in Figure 7.3. On the basis of this we conclude that the WNSF and SLS algorithms is negligible in practice.*

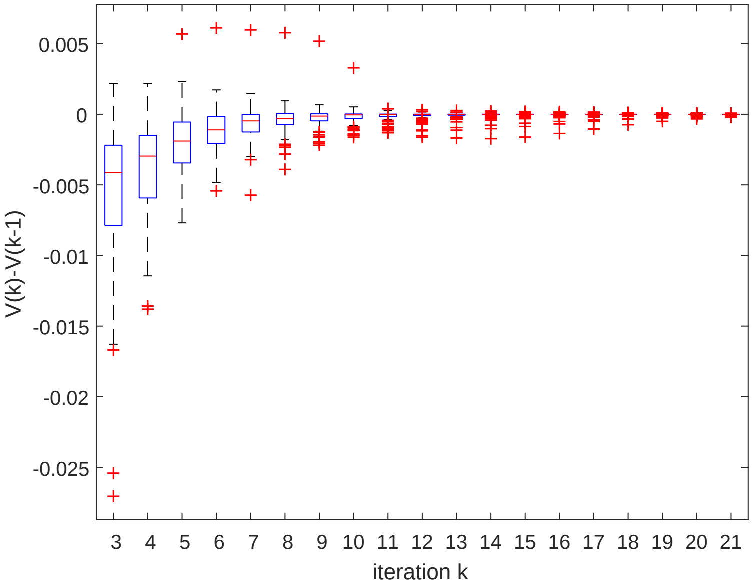


Figure 7.4: Boxplot depicting the improvement of (7.46) over the iterations k. Negative values are an improvement.

We can check convergence of the iterations by evaluating the PEM criterion

$$V(k) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon^T(t, \hat{\theta}^{[k]})\ Q\ \varepsilon(t, \hat{\theta}^{[k]}) \tag{7.46}$$

for each iteration k. Step 2 of Algorithm 7.1 is indicated as $k = 2$, and Step 3 and further iterations with $k \geq 3$. Improvement in cost (7.46) is defined with

$$\Delta(k) = V(k) - V(k-1), \tag{7.47}$$

so when $\Delta(k)\ < 0$ then the cost is improving. Results are plotted for $k \geq 3$ in Figure 7.4. In the plot it is observed that in median the cost improves each iteration until convergence.

There are however a few instances where iterations make the model perform worse in criterion (7.46). From experience, this happens particularly when the ARX order is too large. Despite our best effort to select an appropriate ARX order, we can not guarantee that the optimal one is selected. Therefore it seems beneficial to select the iteration that has the best performance using the PEM criterion. In the current situation with $n_A = 35$, the best iteration is selected using (7.46), and compared to the performance of the final iteration in Figure 7.5. The conclusion is that selecting the best iteration leads to a slight improvement of the fit, but the gain is marginal.

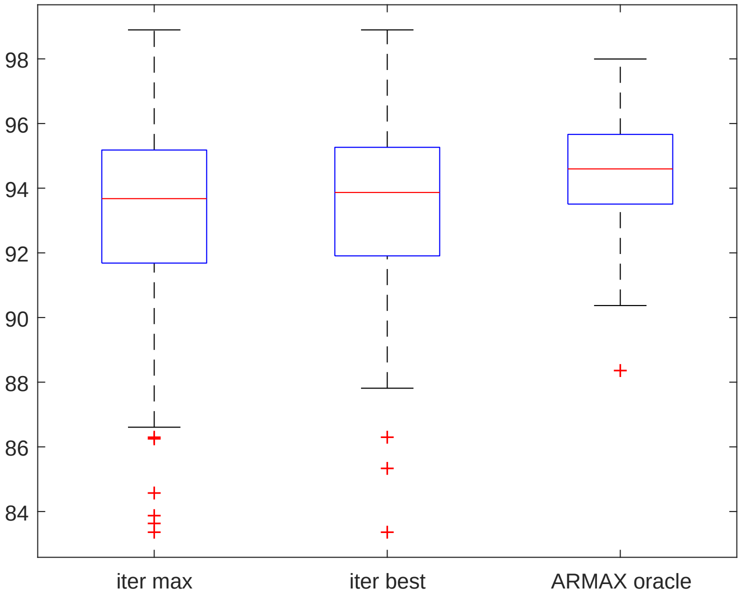


Figure 7.5: Boxplot to compare the final iteration to the iteration selected by criterion (7.46). On the left is the final iteration, in the middle is the best iteration by the criterion, and on the right is the benchmark `ARMAX()` algorithm.

7.4.3 Comparison of methods on a MISO problem

In order to show that the algorithm is competitive with some benchmark algorithms in a multi-input setting we include an estimation of a more challenging 5 node network with randomly generated dynamics. The objective is to estimate G_{12}^0 in the network shown in Figure 7.6 in a MISO setting with a fixed number of samples $N = 1000$. In each Monte-Carlo run, the modules are randomly generated with restrictions:

- Modules are randomly generated by `drss()`. Modules are of 2nd order, with all poles within $|z| < 0.9$ and $\|G(z)\|_{\mathcal{H}'_2} = 0.5$, and all modules on a row of G^0 or H^0 share the same poles.
- The closed-loop transfer $T^0(z)$ and the predictor filters $W^0(z)$ have their poles within $|z| < 0.95$.
- The noise filter H^0 is diagonal and $\Lambda^0 = I$.

The network ARMAX model of order 2 defined in (7.1) is used with the same topology as the network, and with diagonal H. Models are estimated with:

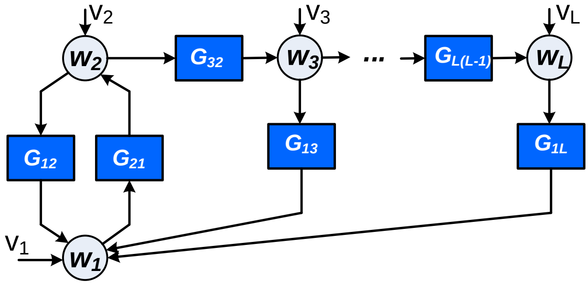


Figure 7.6: An $L = 5$ node network used for simulations.

- Algorithm 7.1 with order and iteration selection.
- The SSARX subspace identification algorithm implemented in Matlab as part of the `n4sid()` function.
- PEM, with the `armax()` algorithm of Matlab with standard initialization.
- PEM, with the `armax()` algorithm of Matlab with the true system as initialization.

In total 100 Monte-Carlo runs are performed, and the resulting fit of module G_{12} is shown in Fig. 7.7. The PEM algorithm starts to struggle with these 4 inputs, but overall the performance of the 4 algorithms is competitive. We conclude that Algorithm 7.1 is suitable for extension to MISO and MIMO, and therefore suitable for use in dynamic networks. SSARX has a slightly better performance than the other algorithms. But as discussed in the introduction, SSARX can not incorporate the network topology in a MIMO estimation setting and is therefore not suitable for identification of a full network.

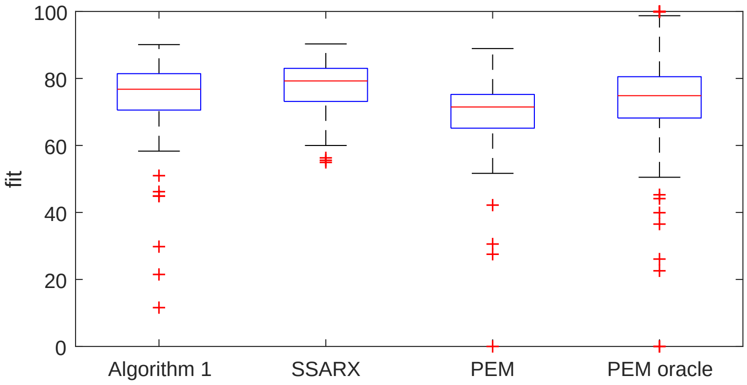


Figure 7.7: Fit of G_{12} for 100 randomly generated systems and data sets for: Algorithm 7.1, SSARX, PEM with standard initialization, PEM initialized by true system.

7.4.4 Estimation with correlated noise

Algorithm 7.1 is presented as a method that can estimate networks where noises are correlated. The network depicted in Figure 7.8 with dynamics

$$G_{12} = \frac{0.3q^{-1} - 0.5q^{-2}}{1 - 0.7q^{-1} + 0.2q^{-2}}, \qquad G_{21} = \frac{0.4q^{-1} - 0.2q^{-2}}{1 - 0.4q^{-1} + 0.8q^{-2}},$$
$$H_{11} = \frac{1 - 0.85q^{-1}}{1 - 0.7q^{-1} + 0.2q^{-2}}, \qquad H_{21} = \frac{0.2q^{-1} + 0.6q^{-2}}{1 - 0.4q^{-1} + 0.8q^{-2}},$$
$$H_{12} = \frac{-0.6q^{-1} + 0.8q^{-2}}{1 - 0.7q^{-1} + 0.2q^{-2}}, \qquad H_{22} = \frac{1 - 0.3q^{-1} + 0.3q^{-2}}{1 - 0.4q^{-1} + 0.8q^{-2}},$$
$$R_{11} = \frac{1}{1 - 0.7q^{-1} + 0.2q^{-2}}, \qquad R_{22} = \frac{1}{1 - 0.4q^{-1} + 0.8q^{-2}},$$

will be simulated such that the estimation performance in the situation of colored and correlated process noise can be tested. Note that in this system the transfer functions on each row have the same denominator, i.e. it is of network ARMAX form. The r_1, r_2, e_1 and e_2 are mutually uncorrelated white noise of equal power. With this network 100 data sets are generated with random noise and external variable realizations.

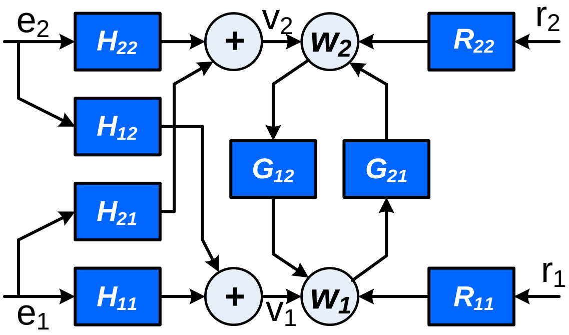


Figure 7.8: Network of 2 nodes with correlated process noise.

Performance of some different estimation methods is tested on the data sets, namely

- Algorithm 7.1 with a network ARMAX model of the same structure as the true network, i.e.

$$M = \left(G = \begin{bmatrix} 0 & G_{12} \\ G_{21} & 0 \end{bmatrix}, R = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix}, H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}, \Lambda = I \right). \quad (7.48)$$

  The ARX model is of order 15 and the estimates obtained in steps $k = 2$ and $k = 3$ are evaluated. The ARMAX model is of order 2.

- The generalized Instrumental Variable method for dynamic networks (Dankers et al., 2015) is used in a 2-input-1-output setting for each node, i.e. $r_1, w_2 \rightarrow w_1$ and $r_2, w_1 \rightarrow w_2$. For this method a MISO ARMAX model of order 2 is used. As instrument both r_1 and r_2 and up to 40 samples of their past are used.

- Direct estimation using the ARMAX() algorithm. The model structure is characterized by

$$\mathcal{A} = \begin{bmatrix} D_{11} & N_{G12} \\ N_{G21} & D_{22} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} N_{R11} & 0 \\ 0 & N_{R22} \end{bmatrix}, \mathcal{C} = \begin{bmatrix} N_{H11} & 0 \\ 0 & N_{H22} \end{bmatrix}, \tag{7.49}$$

  where $\mathcal{A}$ and $\mathcal{C}$ are of 2nd order, and $\mathcal{B}$ contains only a gain. The diagonal C is used for the reason that Matlab's ARMAX() algorithm can not handle non-diagonal C. For this method the network model can not fully capture the true network.

- Direct estimation using the SSARX algorithm is performed by estimating two separate 2-input-1-output models, i.e. $r_1, w_2 \to w_1$ and $r_2, w_1 \to w_2$.

From these methods Algorithm 7.1 and the generalized IV are consistent, while the other two methods are biased due to not properly modeling the noise. The selected model orders lead to the best fit compared to larger and smaller model orders.

The SSARX algorithm can not encode the network topology in a MIMO setting, in particular the 0 entries in R^0 can not be enforced, which would lead to non-identifiability. Therefore the SSARX is applied as two separate MISO identifications, which leads to bias.

The results of the estimation of modules G_{12} and G_{21} are plotted in Figures 7.9a and 7.9b respectively. In these plots the fit of the original transfer function is plotted for each estimation method. As can be expected the two biased methods have poor performance. Generalized IV has acceptable performance, but suffers from a large variance error. Algorithm 7.1 performs well, the asymptotically efficient step 3 shows an improvement over the consistent step 2.

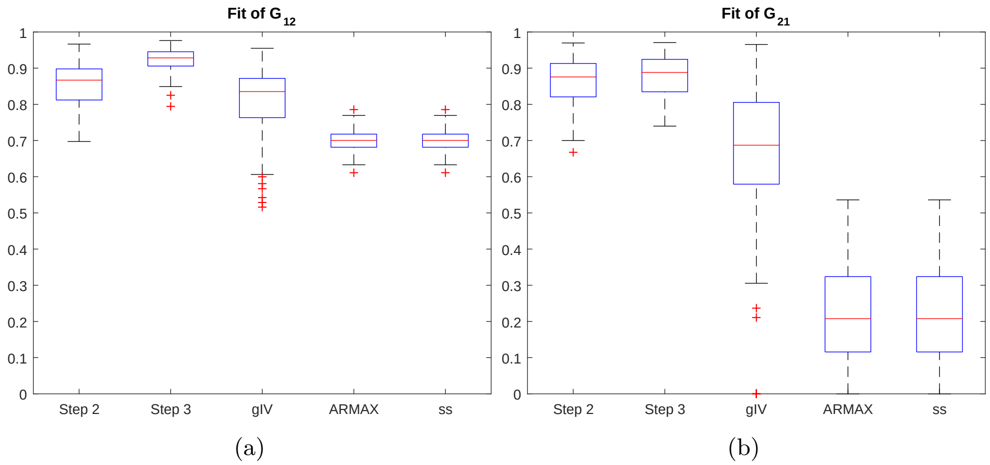
7.5 Conclusions

7.5.1 Conclusions

For the general dynamic network identification problem that includes correlated noises there are a few algorithms available in literature. These algorithms are however not satisfactory in the sense that they do not minimize variance of the estimate. A Sequential Least Squares algorithm is introduced that asymptotically in the number of data points leads to efficient estimates. This algorithm consists of a sequence of least squares problems to be solved, which makes it computationally attractive since there is an explicit solution and local minima are avoided.

7.5.2 Extensions

In Algorithm 7.1 an ARX model is used where each polynomial is of the same order. Systems may best be modeled by different order polynomials for the A and B matrices,



(a)

(b)

Figure 7.9: Fit for module G_{12} (left) and module G_{21} (right). Each boxplot corresponds to one method, from left to right the methods are step 2 and step 3 of Algorithm 7.1, the generalized IV method, direct estimation by the `ARMAX()` algorithm, and the SSARX algorithm.

or even different order polynomials for individual elements of A and B. When the data generating system has different orders for the polynomials, then using the ARX model defined in Algorithm 7.1 introduces either bias or variance error.

When the order of the ARX is the same in every polynomial, then selection over the best order is straightforward to compute. However when for every element of A and B an order must be determined, then order selection becomes computationally heavy. In such a situation one can resort to some heuristic selection algorithm to select individual orders. Or alternatively a regularization scheme can be employed to avoid the order selection problem altogether.

Another possible extension is the use of non-convex optimizations to improve the obtained estimates. The obvious candidate is to use the estimate obtained by Algorithm 7.1 as an initialization to the non-convex joint-direct criterion.

7.6 Appendix

7.6.1 Proof of Proposition 7.3

The proof proceeds by showing that the SLS estimate $W(\hat{\theta}_N^{[k]})$ can be written with the same expression as the WNSF estimate $W(\hat{\theta}_{WNSF}^{[k]})$ for the initial estimate, and after that for the following iterations. From (7.15) the SLS estimator's initial estimate is obtained and restated here

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_{L2}^T(t,\theta)\; Q\; \varepsilon_{L2}(t,\theta) \quad , Q > 0. \tag{7.50}$$

The ε_{L2} expression can after substitution of (7.4) be written as

$$\varepsilon_{L2}(t,\theta) = \mathcal{A}(q,\theta)w(t) - \mathcal{B}(q,\theta)r(t) - \mathcal{C}(q,\theta)\varepsilon_A(t,\hat{\eta}_N). \tag{7.51}$$

When the expression (7.7) for the ε_A is substituted it is obtained that

$$\varepsilon_{L2}(t,\theta) = \begin{bmatrix}\mathcal{A}(q,\theta) - \mathcal{C}(q,\theta)A(q,\hat{\eta}_N) & \mathcal{C}(q,\theta)B(q,\hat{\eta}_N) - \mathcal{B}(q,\theta)\end{bmatrix}\begin{bmatrix}w(t)\\ r(t)\end{bmatrix}. \tag{7.52}$$

Using (7.33) and (7.32) this can be written as

$$\varepsilon_{L2}(t,\theta) = \varepsilon_\eta^T(\eta,\theta)\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}^T\begin{bmatrix}w(t)\\ r(t)\end{bmatrix}. \tag{7.53}$$

Since ε_{L2} is scalar for the SISO case, then the SLS estimate can be written as

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta\in\Theta}\frac{1}{N}\sum_{t=1}^{N}\left(\varepsilon_\eta^T(\eta,\theta)\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}^T\begin{bmatrix}w(t)\\ r(t)\end{bmatrix}\right)^2. \tag{7.54}$$

This can be written as a weighted estimate as

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta\in\Theta}\varepsilon_\eta^T(\eta,\theta)\left(\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}^T\begin{bmatrix}w(t)\\ r(t)\end{bmatrix}\begin{bmatrix}w^T(t) & r^T(t)\end{bmatrix}\begin{bmatrix}\rho & 0\\ 0 & \rho\end{bmatrix}\right)\varepsilon_\eta(\eta,\theta). \tag{7.55}$$

This then shows that the estimate is equal to the WNSF estimate $\hat{\theta}_{WNSF}^{[2]}$ with the weight defined by $W = P$ defined by (7.37).

In the further iterations of the SLS an additional weight is included in the estimator, the new criterion is given by (7.17)

$$\hat{\theta}_N^{[k]} = \arg\min_{\theta\in\Theta}\frac{1}{N}\sum_{t=1}^{N}\varepsilon_{Lk}^T(t,\theta)\; Q\; \varepsilon_{Lk}(t,\theta), \tag{7.56}$$

with

$$\varepsilon_{Lk}(t,\theta) := N_H^{-1}(q,\hat{\theta}_N^{[k-1]})\varepsilon_{L(k-1)}(t,\theta). \tag{7.57}$$

Now the linear regression can be extended with an additional noise model, and to this end (7.33) is extended. The linear regression $\varepsilon_\eta(\eta,\theta)$ is then pre-multiplied with

$$\begin{bmatrix}\mathcal{T}(c) & 0\\ 0 & \mathcal{T}(c)\end{bmatrix}, \tag{7.58}$$

where c is a column vector containing the impulse response of $\mathcal{C}^{-1}(q,\theta_C^{[k-1]})$, implying that the dimension of $\mathcal{T}(c)$ is $\infty\times(n_A+n_p+1)$. Then the filter of ε_{Lk} can be written as

$$\begin{aligned}\varepsilon_\eta^T(\eta,\theta)\begin{bmatrix}\mathcal{T}(c) & 0\\ 0 & \mathcal{T}(c)\end{bmatrix}^T\begin{bmatrix}\rho^\infty & 0\\ 0 & \rho^\infty\end{bmatrix} = \cdot\\ \cdot\,\mathcal{C}^{-1}(q,\theta_C^{[k-1]})\begin{bmatrix}\mathcal{C}(q,\theta_C)A(q,\eta_A) - \mathcal{A}(q,\theta_A) & \mathcal{C}(q,\theta_C)B(q,\eta_B) - \mathcal{B}(q,\theta_B)\end{bmatrix},\end{aligned} \tag{7.59}$$

where $\rho^\infty = \begin{bmatrix} 1 & q^{-1} & \cdots & q^{-\infty} \end{bmatrix}^T$.

An approximation can be made by truncating the impulse response of $\mathcal{C}^{-1}(q, \theta_C^{[k-1]})$ at $(n_A + n_p + 1)$ parameters, i.e.

$$\begin{bmatrix} \mathcal{T}(c) & 0 \\ 0 & \mathcal{T}(c) \end{bmatrix} \approx \begin{bmatrix} \mathcal{T}\left(\begin{bmatrix} 1 \\ \theta_C^{[k-1]} \\ 0_{n_A \times 1} \end{bmatrix}\right) & 0 \\ 0 & \mathcal{T}\left(\begin{bmatrix} 1 \\ \theta_C^{[k-1]} \\ 0_{n_A \times 1} \end{bmatrix}\right) \end{bmatrix}^{-1} = T_C^{-1}(\theta_C^{[k-1]}), \tag{7.60}$$

where the dimension of T_C is $2(n_A + n_p + 1) \times 2(n_A + n_p + 1)$. Then the filter is

$$\begin{aligned} &\varepsilon_\eta^T(\eta, \theta) T_C^{-T}(\theta_C^{[k-1]}) \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix} = \cdot \\ &\cdot \mathcal{C}^{-1}(q, \theta_C^{[k-1]}) \begin{bmatrix} \mathcal{C}(q, \theta_C) A(q, \eta_A) - \mathcal{A}(q, \theta_A) & \mathcal{C}(q, \theta_C) B(q, \eta_B) - \mathcal{B}(q, \theta_B) \end{bmatrix}, \end{aligned} \tag{7.61}$$

For the estimation the following is then obtained

$$\hat{\theta}_N^{[k]} = \arg\min_{\theta \in \Theta} \varepsilon_\eta^T(\eta, \theta) T_C^{-T}(\theta_C^{[k-1]}) P T_C^{-1}(\theta_C^{[k-1]}) \varepsilon_\eta(\eta, \theta), \tag{7.62}$$

where

$$P = \left(\frac{1}{N} \sum_{t=1}^{N} \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix}^T \begin{bmatrix} w(t) \\ r(t) \end{bmatrix} \begin{bmatrix} w^T(t) & r^T(t) \end{bmatrix} \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix} \right), \tag{7.63}$$

which is exactly the WNSF estimator $\hat{\theta}_{WNSF}^{[k]}$. ■

8 Conclusions

8.1 Answer to the research question

The need to reduce greenhouse gas emissions, modernization of the electricity grid, and other engineering challenges have created a need to further advance knowledge of dynamical systems. For various purposes such as analysis of a system or control design there is a need for high quality models of the dynamical behavior as well as the internal structure of the system. Modern systems often consist of multiple interconnected subsystems, which can be modeled as a dynamic network. The research question answered in this thesis is under which conditions dynamic network models can efficiently be estimated. A discussion of the main contributions to answering the research question is provided next.

Network identifiability

Contrary to open-loop and closed-loop system, the experimental setup of a dynamic network is flexible, and multiple network topologies may describe the same dynamical system. The original intention of the identifiability study has been to find conditions under which different network topologies can be distinguished in a topology detection problem. It quickly became clear that the presence and modeled location of external signals is critical to be able to distinguish between different network topologies. This eventually led to conditions on the modeled experimental setup, which need to be satisfied in order to distinguish between different networks.

The main concept of network identifiability has been introduced as a concept that tells us that network models can be distinguished from each other. Network identifiability is a property of a set of network models that describes that module dynamics and topology are uniquely represented within the model set, which makes it a different type of identifiability that is not focused on parameters. Network identifiability is a necessary condition for the consistent estimation of networks, and conditions under which networks are identifiable have been obtained. Requiring that every node has an independent external excitation or noise source is sufficient to guarantee network identifiability, but also conservative. Non-conservative conditions have been derived, which allows us to check the identifiability of any dynamic network model set. Relaxed conditions for the identifiability of just a single module have been formulated, which become relevant when the objective is to identify only a single module. The conditions for identifiability can generically be checked on the basis the modeled network topology.

Any traditional MIMO system can be modeled as a dynamic network, and the user can choose the modeled network structure regardless of the structure of the data generating network. The user can select a structure that either matches the physical structure, that is beneficial for identification, or beneficial for control purposes. In fact a traditional MIMO model is one special case of the dynamic network model.

Joint-direct method

An investigation into algebraic loops triggered the realization that we need to consider MIMO predictors. It then became clear that confounding variables can be modeled as correlated disturbances and included in the predictor. The main identification method introduced in this thesis, the joint-direct method, has been the natural follow-up of the MIMO predictor.

The joint-direct method is an asymptotically efficient identification method, which is applicable to any experimental setup. In particular the joint-direct method can identify networks where noises are correlated and rank-reduced, and it can identify networks that contain algebraic loops. Correlated disturbances are taken care of by jointly predicting all node signals, and modeling a multivariable noise model. Rank-reduced noises may be modeled with non-square noise models, and estimated with a constrained criterion, which can be relaxed to an unconstrained criterion. Algebraic loops are also taken care of by jointly predicting all node signals, and by making explicit use of external excitations to uniquely determine all the feedthrough terms. All these situations lead to consistency and asymptotic efficiency under certain conditions. One of the main conditions for consistency of the joint-direct method is network identifiability. Another important condition, the informativity of data, i.e. the sufficient excitation of all relevant dynamics, has been assumed to hold, but this requires further investigation. The variance of estimates obtained with the joint-direct method asymptotically reaches the Cramér-Rao lower bound. In particular for the rank-reduced noise situation, the Cramér-Rao lower bound has been reformulated in order to show that this bound is reached.

In practical situations it is easy to come up with reasons why noises are correlated, for example a wind disturbance that affects multiple position measurements. Network models for such practical scenario's can be identified with the joint-direct method, while many other methods in literature are not able to handle these practical situations in a satisfactory way.

For estimation problems that involve algebraic loops, typically indirect methods are formulated that do not minimize the variance. A direct method for the situation that noise is full rank has been formulated. This method makes use of external excitations only to provide uniqueness of the feedthrough terms involved in the algebraic loop, and not to de-correlate noises. Utilizing the excitation provided by process noise is the main reason why the joint-direct method improves the variance compared to indirect methods.

Single module identification

In practical situations one may need only part of a dynamic network model, for example to analyze the local behavior or develop a local controller. For this reason it is interesting to be able to identify part of a network, or a single module in particular. To model a single module we typically do not need to measure all nodes. The problem of which nodes need to be included in a network description such that the module of interest can be identified has been investigated on the basis of the concept of immersion. Immersion is a method that removes nodes from a network description while leaving the other nodes and dynamics invariant. When the module of interest is left invariant after immersion, then the removed nodes are not needed for consistent identification of the module.

It turns out that immersion is just one way of removing nodes from a network description, as there is also the indirect inputs method that can remove nodes. This has lead to the notion of abstraction, which is a generalization of the immersion method and the indirect inputs method. In this way an abstracted network description can be determined that includes only a selected set of nodes. Conditions under which abstraction leaves the dynamics of a module of interest invariant, and conditions under which this module is also identifiable in the abstracted network have been obtained. Under those conditions the module of interest may be identified on the basis of the selected nodes. An insight obtained is that not only a set of input nodes must be selected in the local modeling procedure, but also a set of output nodes must be selected.

Sequential Least Squares algorithm

Efficient algorithms are required in order to bring network identification techniques into practice. The joint-direct method is formulated for simultaneous identification of all modules in a network by a typically non-convex cost function. In particular for large-scale networks it is desirable to avoid non-convex optimization schemes. Being able to encode the network topology into the algorithm is required for the joint-direct method, which is not possible in typical identification algorithms.

An algorithm based on a sequential application of least squares estimations has been introduced as an approximation of the joint-direct method. This algorithm can encode the network topology, and this includes the correlation structure of noises. Moreover, the algorithm leads to asymptotically efficient estimates while avoiding local minima. Since every step of the algorithm is based on linear regression, it is expected that this algorithm can scale well to large-scale networks.

The Sequential Least Squares algorithm is suited for identification of a part of a network, or of a single module. For example a network identification problem where noises are uncorrelated can be split into a MISO identification problem for each node, and this is straightforward to identify with the algorithm.

Global research question

Reflecting back onto the global research question, under certain conditions it is possible to efficiently estimate all modules in a network with prediction error methods. The main conditions are that a network model set is network identifiable, which is essentially a condition on the modeled network topology, and the data should contain sufficient information. Identifying a particular module of interest is also possible under some conditions on the network topology, although there is still much work to be done in this identification problem. A solid theoretical basis has been formed that may be extended to the identification of single modules, the identification of topology, or other problems.

8.2 Recommendations for future research

8.2.1 Informativity of data

A limitation of the theory presented in this thesis is that the data has been treated as if it is informative, which has allowed us to avoid investigating informativity of data in detail. In the joint-direct method some nodes may be used both as input and as output simultaneously, which leads to the fundamental question what data informativity is for dynamic networks, and how to check it. Analysis of informativity is connected to the identifiability analysis that has been carried out, and that is likely to be a good starting point for future research. Informativity for identification of local modules is an interesting topic in the sense that conditions can be relaxed considerably compared to the full network situation, as only the module of interest needs to be uniquely identified and therefore the only module that needs to be sufficiently excited.

When the requirements for informativity have been determined, then this topic may be continued with an investigation into experiment design. Where to excite the network such that variance of a module, or modules, is minimized is a question that remains open.

8.2.2 Topology detection

Identification of a dynamic network under the assumption that the interconnection structure is unknown is a possible extension of the problem setting of this thesis. Bayesian estimation techniques are a promising direction that may be able to identify the interconnection structure along with the network dynamics. The Bayesian identification techniques are related to regularized identification techniques that use impulse response models for the dynamics. Introducing dynamic network models with colored and correlated noise in the Bayesian estimation problem may lead to topology detection schemes that function under rather general assumptions on the experimental setup. An additional benefit of the regularized identification techniques is that the model order of modules does not have to be chosen, as the modules are modeled as impulse responses.

8.2.3 Non-linear models

Only linear models and systems have been treated in this thesis. Some common non-linear phenomena such as static friction or saturation can not be captured by linear models. An extension that is relevant for practical identification problems is the inclusion of non-linear dynamics into the modules. Adding non-linear dynamics can be achieved for example by multiplying each linear module with a static non-linearity, although other approaches such a linear-parameter-varying models may be just as valid. Due to the ability of the network to keep dynamics induced by a non-linear component localized around that component, the non-linearities may be added to only a few modules, such that the identification problem is kept relatively simple. A question that then arises is where to model the non-linearities, and whether modules can remain as being single-input-single-output.

8.2.4 Local identification

The research topic of identifying a single module in a network is far from exhausted. In literature on single module identification the focus has been on choosing the appropriate inputs, but one of the realizations from this thesis is that additional outputs may need to be modeled. Thorough investigation of which nodes are to be used as inputs and/or outputs, how the abstracted network structure is modeled, and what kind of information needs to be present in data is required in future research.

A requirement imposed in this thesis has been that the module of interest remains invariant under abstraction, but that is not a strict requirement for consistent identification. It may be that the module of interest can be identified in an indirect way by combining the knowledge of two or more modules present in the abstracted network. This points to the question whether a module is identifiable on the basis of a subset of node signals.

Bibliography

K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5): 551–574, 1980.

J. Bang-Jensen and G. Z. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer Science & Business Media, 2008.

A. S. Bazanella, M. Gevers, and L. Miskovic. Closed-loop identification of MIMO systems: a new look at identifiability and experiment design. *European Journal of Control*, 16(3):228–239, May–June 2010.

A.S. Bazanella, M. Gevers, J. Hendrickx, and A. Parraga. Identifiability of dynamical networks: which nodes need be measured? In *Proc. 56th IEEE Conf. Decision and Control (CDC 2017)*, pages 5870–5875, 2017.

H. P. G. J. Beelen, L. H. J. Raijmakers, M. C. F. Donkers, P. H. L. Notten, and H.J. Bergveld. A comparison and accuracy analysis of impedance-based temperature estimation methods for li-ion batteries. *Applied Energy*, 175:128–140, 2016.

K. A. Bollen. *Structural Equations with Latent Variables.* John Wiley & Sons, 1989.

G. Bottegal, A. Chiuso, and P. M. J. Van den Hof. On dynamic network modeling of stationary multivariate processes. In *Prepr. 18th IFAC Symposium on System Identification, Stockholm, Sweden.* IFAC, 2018. (To appear).

P. E. Caines and C. W. Chan. Feedback between stationary stochastic processes. *IEEE Trans. on Automatic Control*, 20(4):498–508, 1975.

A. Chiuso and G. Pillonetto. A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553–1565, 2012.

J. Cook, D. Nuccitelli, S. A. Green, M. Richardson, B. Winkler, R. Painting, R. Way, P. Jacobs, and A. Skuce. Quantifying the consensus on anthropogenic global warming in the scientific literature. *Environmental research letters*, 8(2):024024, 2013.

A. G. Dankers. *System Identification in Dynamic Networks.* PhD thesis, Delft University of Technology, 2014.

A. G. Dankers, P. M. J. Van den Hof, X. Bombois, and P. S. C. Heuberger. Errors-in-variables identification in dynamic networks - Consistency results for an instrumental variable approach. *Automatica*, 62:39–50, 2015.

A. G. Dankers, P. M. J. Van den Hof, P. S. C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods: Predictor input selection. *IEEE Trans. on Automatic Control*, 61(4):937–952, 2016.

A. G. Dankers, P. M. J. Van den Hof, D. Materassi, and H. H. M. Weerts. Conditions for handling confounding variables in dynamic networks. In *Proc. 20th IFAC World Congress*. IFAC, 2017. (IFAC-PapersOnLine, 50(1):3983-3988).

M. Deistler, W. Scherrer, and B. D. O. Anderson. The structure of generalized linear dynamic factor models. In J. Beran, Y. Feng, and H. Hebbel, editors, *Empirical Economic and Financial Research*, pages 379–400. Springer, 2015.

N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Connaught Circus, New Delhi: Prentice hall of India, 1974.

J.-M. Dufour and T. Jouini. Asymptotic distributions for quasi-efficient estimators in echelon VARMA models. *Computational Statistics & Data Analysis*, 73:69–86, 2014.

J. Durbin. The fitting of time-series models. *Revue de l'Institut International de Statistique*, 28(3):233–244, 1960.

European Technology Platform SmartGrids. *Vision and Strategy for Europe's Electricity Networks of the Future*. European Commision Community Research, 2006.

N. Everitt. *Module identification in dynamic networks: parametric and empirical Bayes methods*. PhD dissertation, KTH Royal Institute of Technology, 2017.

N. Everitt, C. R. Rojas, and H. Hjalmarsson. A geometric approach to variance analysis of cascaded systems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conf. on*, pages 6496–6501. IEEE, 2013.

N. Everitt, G. Bottegal, C. R. Rojas, and H. Hjalmarsson. On the effect of noise correlation in parameter identification of simo systems. In *Proc. 17th IFAC Symposium on System Identification*. IFAC, 2015. (IFAC-PapersOnLine, 48(28):326-331).

N. Everitt, G. Bottegal, and H. Hjalmarsson. An empirical Bayes approach to identification of modules in dynamic networks. *Automatica*, 91:144 – 151, 2018.

P. Eykhoff. *System identification: parameter and state estimation*. Wiley, London, 1974.

E. Felsenstein. *Regular and Singular AR and ARMA Models: The Single and The Mixed Frequency Case*. PhD dissertation, Vienna University of Technology, 2014.

M. Galrinho. *Least Squares Methods for System Identification of Structured Models*. Licentiate thesis, KTH Royal Institute of Technology, 2016.

M. Galrinho, C.R. Rojas, and H. Hjalmarsson. A weighted least-squares method for parameter estimation in structured models. In *53rd IEEE Conference on Decision and Control*, pages 3322–3327, Los Angeles, USA, 2014.

M. Galrinho, C.R. Rojas, and H. Hjalmarsson. Parametric identification using weighted null-space fitting. *arXiv preprint arXiv:1708.03946*, 2017.

M. Gevers, A.S. Bazanella, and A. Parraga. Structural conditions for the identifiability of dynamical networks. In *Preprints 20th IFAC World Congress*, pages 11069–11074, 2017.

M. Gilson and P. M. J. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2):241–249, 2005.

J. Gonçalves and S. Warnick. Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Trans. Automatic Control*, 53(7):1670–1674, August 2008.

C. W. J. Granger. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

J. L. Gross and J. Yellen. *Graph theory and its applications*. CRC press, 2005.

A. Haber and M. Verhaegen. Subspace identification of large-scale interconnected systems. *IEEE Transactions on Automatic Control*, 59(10):2754–2759, 2014.

E. J. Hannan and M. Deistler. *The Statistical Theory of Linear Systems.* John Wiley & Sons, New York, 1988.

E. J. Hannan and L. Kavalieris. Linear estimation of ARMA processes. *Automatica*, 19(4):447–448, 1983.

N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay. Microgrids. *IEEE power and energy magazine*, 5(4):78–94, 2007.

J. M. Hendrickx, M. Gevers, and A. S. Bazanella. Identifiability of dynamical networks with partial node measurements. *ArXiv e-prints*, 2018. arXiv:1803.05885 [math.OC].

B. L. Ho and R. E. Kálmán. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

Intergovernmental Panel on Climate Change. *Climate change 2014: mitigation of climate change*, volume 3. Cambridge University Press, 2015.

M. Jansson. Subspace identification and arx modeling. In *proc. 13th IFAC Symp on System Identification, Rotterdam, The Netherlands*, pages 1585–1590, 2003.

C.G. Khatri. Some results for the singular normal multivariate regression models. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 267–280, 1968.

L. Kölbl. *VAR Systems: g-Identifiability and Asymptotic Properties of Parameter Estimates for the Mixed-Frequency Case.* PhD dissertation, Vienna University of Technology, 2015.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

P. Kundur, N. J. Balu, and M. G. Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.

J. Linder. *Indirect system identification for unknown input problems with applications to ships.* PhD dissertation, Linköping University, 2017.

J. Linder and M. Enqvist. Identification of systems with unknown inputs using indirect input measurements. *International Journal of Control*, 90(4):729–745, 2017a.

J. Linder and M. Enqvist. Identification and prediction in dynamic networks with unobservable nodes. *IFAC-PapersOnLine*, 50(1):10574–10579, 2017b.

L. Ljung. On the consistency of prediction error identification methods. In R. K. Mehra and D. G. Lainiotis, editors, *System Identification: Advances and Case Studies*, pages 121–164. Academic Press, New York, 1976.

L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, NJ, 1999.

P. Massioni. Distributed control for alpha-heterogeneous dynamically coupled systems. *Systems & Control Letters*, 72:30–35, 2014.

D. Materassi and G. Innocenti. Topological identification in networks of dynamical systems. *IEEE Trans. on Automatic Control*, 55(8):1860–1871, 2010.

D. Materassi and M. Salapaka. On the problem of reconstructing an unknown topology via locality properties of the Wiener filter. *IEEE Trans. on Automatic Control*, 57 (7):1765–1777, 2012.

A. Mauroy and J. M. Hendrickx. Spectral identification of networks with inputs. In *Proc. 56th IEEE Conference on Decision and Control (CDC)*, pages 469–474. IEEE, 2017.

D.Q. Mayne and F. Firoozan. Linear identification of ARMA processes. *Automatica*, 18:461–466, 1982.

D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, et al. Trends in microgrid control. *IEEE Transactions on smart grid*, 5(4):1905–1919, 2014.

J. Pearl. *Causality.* Cambridge University Press, 2009.

C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–91, 1945.

C. R. Rao. *Linear statistical inference and its applications.* Wiley, London, 2nd edition, 1973.

C. R. Rojas and H. Hjalmarsson. Sparse estimation based on a validation criterion. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2825–2830. IEEE, 2011.

B. M. Sanandaji, T. L. Vincent, and M. B. Wakin. Exact topology identification of large-scale interconnected dynamical systems from compressive observations. In *American Control Conference*, pages 649–656, San Francisco, CA, USA, 2011.

B. M. Sanandaji, T. L. Vincent, and M. B. Wakin. A review of sufficient conditions for structure identification in interconnected systems. In *16th IFAC Symposium on System Identification*, pages 1623–1628, 2012.

T. Söderström. *Errors-in-variables methods in system identification.* Springer, 2018.

T. Söderström and P. Stoica. *System Identification.* Prentice Hall, Hemel Hempstead, UK, 1989.

T. Söderström, L. Ljung, and I. Gustavsson. Identifiability conditions for linear multivariable systems operating under feedback. *Automatic Control, IEEE Trans. on*, 21(6):837–840, 1976.

M. S. Srivastava and D. von Rosen. Regression models with unknown singular covariance matrix. *Linear algebra and its applications*, 354(1):255–273, 2002.

P. Stoica and B.C. Ng. On the cramér-rao bound under parametric constraints. *IEEE Signal Processing Letters*, 5(7):177–179, 1998.

P. Torres, J. W. van Wingerden, and M. Verhaegen. Output-error identification of large scale 1d-spatially varying interconnected systems. *IEEE Trans. on Automatic Control*, 60(1):130–142, 2015.

United Nations. *Paris Agreement*, chapter TREATIES-XXVII.7.d. United Nations Treaty Collection, 2016.

P. M. J. Van den Hof and R. J. P. Schrama. An indirect method for transfer function estimation from closed loop data. *Automatica*, 29(6):1523–1527, 1993.

P. M. J. Van den Hof, A. G. Dankers, P. S. C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods - basic methods for consistent module estimates. *Automatica*, 49(10):2994–3006, 2013.

P. M. J. Van den Hof, A. G. Dankers, and H. H. M. Weerts. From closed-loop identification to dynamic networks: generalization of the direct method. In *Proc. 56th IEEE Conference on Decision and Control (CDC)*, pages 5845–5850. IEEE, 2017a.

P. M. J. Van den Hof, H. H. M. Weerts, and A.G. Dankers. Prediction error identification with rank-reduced output noise. In *Proc. 2017 American Control Conference*, pages 382–387, Seattle, Florida, 2017b.

J.W van der Woude. A graph-theoreric characterization for the rank of the transfer matrix of a structured system. *Mathematics of Control, Signals, and Systems*, 4: 33–40, 1991.

P. Van Overschee and B. L. De Moor. *Subspace identification for linear systems: Theory–Implementation–Applications.* Springer Science & Business Media, 2012.

H. H. M. Weerts, A. G. Dankers, and P. M. J. Van den Hof. Identifiability in dynamic network identification. In *Prepr. 17th IFAC Symposium on System Identification, Beijing, China*, pages 1409–1414, 2015. (IFAC-PapersOnLine, 48(28):1409-1414).

H. H. M. Weerts, A. G. Dankers, and P. M. J. Van den Hof. Identifiability of dynamic networks with part of the nodes noise-free. Technical report, Eindhoven University of Technology, `http://pvandenhof.nl/Reportfiles/Weerts&etal_ALCOSP2016_report.pdf`, 2016a.

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identification of dynamic networks operating in the presence of algebraic loops. In *Proc. 55nd IEEE Conference on Decision and Control*, pages 4606–4611, 2016b.

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identifiability of dynamic networks with part of the nodes noise-free. In *Proc. 12th IFAC Workshop ALCOSP*, 2016c. (IFAC-PapersOnLine, 49(13):19-24).

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identification of dynamic networks with rank-reduced process noise. In *Proc. 20th IFAC World Congress.* IFAC, 2017. (IFAC-PapersOnLine, 50(1):10562-10567).

H. H. M. Weerts, M. Galrinho, G. Bottegal, H. Hjalmarsson, and P. M. J. Van den Hof. A sequential least squares algorithm for armax dynamic network identification. In *Prepr. 18th IFAC Symposium on System Identification, Stockholm, Sweden.* IFAC, 2018a. pages 844–849.

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Single module identifiability in linear dynamic networks. *arXiv preprint arXiv:1803.02586*, 2018b. (Accepted for presentation at CDC 2018).

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identifiability of linear dynamic networks. *Automatica*, 89:247–258, 2018c.

H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identification of linear dynamic networks with rank-reduced noise. *Automatica*, 98:256–268, 2018d.

J. C. Willems. *Analysis of feedback systems.* Number 62 in Research Monograph. Cambridge, Massachussets: The M.I.T. Press., 1971.

J. C. Willems and J. W. Polderman. *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media, 2013.

D.C. Youla. On the factorization of rational matrices. *IRE Trans. Information Theory*, 7:172–189, 1961.

Y. Yuan. *Decentralised Network Prediction and Reconstruction Algorithms.* PhD dissertation, University of Cambridge, 2012.

Y. Yuan, G. B. Stan, S. Warnick, and J. Gonçalves. Robust dynamical network structure reconstruction. *Automatica*, 47(6):1230–1235, 2011.

Z. Yue, J. Thunberg, W. Pan, L. Ljung, and J. Gonçalves. Linear dynamic network reconstruction from heterogeneous datasets. *IFAC-PapersOnLine*, 50(1):10586–10591, 2017.

M. Zorzi and A. Chiuso. Sparse plus low rank network identification: A nonparametric approach. *Automatica*, 76:355–366, 2017.

List of Abbreviations

AIC	Akaike's Information Criterion
AR	AutoRegressive
ARMA	AutoRegressive Moving Average
ARMAX	AutoRegressive Moving Average with eXogeneous input
ARX	AutoRegressive with eXogeneous input
BJ	Box-Jenkins
CLS	Constrained Least Squares
CRLB	Cramér-Rao lower bound
DSF	Dynamic Structure Function
FIR	Finite Impulse Response
IV	Instrumental Variable
LTI	Linear Time-Invariant
MIMO	Multi-Input-Multi-Output
MISO	Multi-Input-Single-Output
ML	Maximum Likelihood
MORSM	Model Order Reduction Steiglitz-McBride
MSE	Mean Squared Error
OE	Output Error
PEM	Prediction error method
SISO	Single-Input-Single-Output
SLS	Sequential Least Squares
WLS	Weighted Least Squares
WNSF	Weighted NullSpace Fitting

Acknowledgements

The past four years have passed very quickly and during this time I met many inspiring people. Without the efforts of these people this thesis would not have been completed.

First of all I would like to thank my supervisor and promotor Prof. Paul Van den Hof for all the guidance and support I have received during the time we worked together. My initial objective when starting the PhD was to challenge my capabilities, and the many in-depth discussions that we had have really fulfilled this objective. I also feel like I learned a lot from you about logically building a story to explain my thoughts, and on communication skills in general.

Secondly I will express my gratitute towards my co-promotor Arne Dankers who has been supervising me during the master thesis, and during the PhD trajectory. I really appreciate the freedom that you have given me to develop my own thoughts and to build a reasoning from that. It was always nice to read your comments and emails due to the positive attitude that you transmit with them.

I extend my thanks to the members of my committee Prof. Jorge Gonçalves, Prof. Jacquelien Scherpen, Prof. Natal van Riel, Prof. Siep Weiland, and Prof. Martin Enqvist. Your efforts in evaluating my work are appreciated, and your comments have improved the quality of the thesis.

My time in the Control Systems group has been an enjoyable experience due to the wonderful atmosphere. Thank you to all my office mates for providing a nice working environment, it wouldn't have been the same without you. I also enjoyed playing futsal with 'The team of Rob', later renamed to 'Atletico Bitterballen'. Thanks Koen, Veaceslav, Ryan, Tuan, Hernan, Pepijn, Mohammed, Mircea, Henrik, Giuseppe, Ioannis, Alejandro, Giulio and everyone else that played with us. Surely the team will enter an era of greatness after my retirement. The PhD life also involved traveling to international conferences which can be intimidating if you do not know anyone, but luckily I have fond memories of traveling with Constantijn, Giuseppe, Tuan, Dhruv, Ruxandra, Pepijn, Sofie, Amritam, Maarten, Ruben, Marcella, Daming etc. The support and atmosphere provided by Barbara, Diana and Hiltje are also great.

Taking my mind off of work was an important task taken care of by my friends. Traveling together was great, drinking beers together was great, and playing games was great, thanks! Near the end of the writing phase I should have listened to your advice to take more breaks.

I want to thank my parents Henk and Hanny for always letting me make my own choices, and for supporting me through my journey throughout the university. Also my sisters Marianne, Annette and Ingrid, thanks for motivating me to give my best. It was very helpful for me to try and explain my work in 'human language'. A special thanks goes out to Ingrid for designing a wonderful cover.

Ik wil mijn ouders Henk en Hanny bedanken voor de vrije keuzes die ze me altijd lieten maken, en voor de steun tijdens mijn reis in de universiteit. Ook mijn zussen Marianne, Annette en Ingrid, bedankt voor de motivatie om mijn beste beentje voor te zetten. Het was heel nuttig om te proberen mijn werk uit te leggen in 'mensentaal'. Een speciaal bedankje is er voor Ingrid vanwege het ontwerpen van de prachtige kaft.

Curriculum Vitae

Harm Weerts was born in Bergen, The Netherlands, in 1989. He received his Bachelor degree from the Fontys University of Applied Sciences in Venlo, the Netherlands, in 2010. Afterwards he started working towards the Master degree Systems & Control at Eindhoven University of Technology, from which he graduated in 2014. During his master degree studies he did a 3 month internship in Aalborg University, Denmark, on demand response using refrigeration systems. His M.Sc. thesis was titled ”Topology detection in dynamic networks”. In 2014 he was appointed as a PhD candidate in the Department of Electrical Engineering at Eindhoven University of Technology. During the period 2014-2016 he took graduate courses at the Dutch Institute for Systems and Control (DISC) and received the DISC certificate. His Ph.D. research is on system identification applied to dynamic networks with a multivariable approach, of which the results are presented in this thesis.