Linköping Studies in Science and Technology. Dissertations.
No. 1046

Identification and Estimation for Models Described by Differential-Algebraic Equations

Markus Gerdin

Department of Electrical Engineering
Linköpings universitet, SE–581 83 Linköping, Sweden

Linköping 2006

Identification and Estimation for Models Described by Differential-Algebraic Equations

© 2006 Markus Gerdin

gerdin@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet
SE–581 83 Linköping
Sweden

ISBN 91-85643-87-4 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2006

To Jessica

Abstract

Differential-algebraic equations (DAEs) form the natural way in which models of physical systems are delivered from an object-oriented modeling tool like Modelica. Differential-algebraic equations are also known as descriptor systems, singular systems, and implicit systems. If some constant parameters in such models are unknown, one might need to estimate them from measured data from the modeled system. This is a form of system identification called gray box identification. It may also be of interest to estimate the value of time-varying variables in the model. This is often referred to as state estimation. The objective of this work is to examine how gray box identification and estimation of time-varying variables can be performed for models described by differential-algebraic equations.

If a model has external stimuli that are not measured or uncertain measurements, it is often appropriate to model this as stochastic processes. This is called noise modeling. Noise modeling is an important part of system identification and state estimation, so we examine how well-posedness of noise models for differential-algebraic equations can be characterized. For well-posed models, we then discuss how particle filters can be implemented for estimation of time-varying variables. We also discuss how constant parameters can be estimated.

When estimating time-varying variables, it is of interest to examine if the problem is observable, that is, if it has a unique solution. The corresponding property when estimating constant parameters is identifiability. In this thesis, we discuss how observability and identifiability can be determined for DAEs. We propose three approaches, where one can be seen as an extension of standard methods for state-space systems based on rank tests.

For linear DAEs, a more detailed analysis is performed. We use some well-known canonical forms to examine well-posedness of noise models and to implement estimation of time-varying variables and constant parameters. This includes formulation of Kalman filters for linear DAE models. To be able to implement the suggested methods, we show how the canonical forms can be computed using numerical software from the linear algebra package LAPACK.

Acknowledgments

There are several people who helped me during the work with this thesis. First of all I would like to thank my supervisors Professor Torkel Glad and Professor Lennart Ljung for guiding me in my research in an excellent way and always taking time to answer my questions. It has been a privilege to have you both by my side during my time as a Ph.D. student.

Furthermore, I would like to thank everyone at the Control and Communication group for providing a nice working atmosphere. I am going to miss the coffee room discussions. I would like to mention Johan Sjöberg for many enlightening discussions on DAE models and the cooperation on noise modeling for nonlinear DAE models, and Dr. Thomas Schön for the cooperation on the work on noise modeling for linear DAE. This thesis has been proofread by Gustaf Hendeby, Dr. Jacob Roll, Dr. Thomas Schön, Johan Sjöberg, and Henrik Tidefelt. You all helped improve the quality of the thesis. I also thank Ulla Salaneck for helping with many practical issues, always with a cheerful attitude.

This work has been supported by the Swedish Foundation for Strategic Research (SSF) through VISIMOD and ECSEL and by the Swedish Research Council (VR) which is gratefully acknowledged.

Finally I would like to thank my family and friends for inspiration and support. You are important to me, even though I might have neglected you during the work with this thesis. Finally I thank Jessica for understanding when I had to focus on writing this thesis. You are an important part of my life.

Contents

1 Introduction **1**
1.1 Problem Formulation 2
1.2 Outline 4
1.3 Contributions 4

2 Modeling **7**
2.1 Introduction: Component-Based Modeling 7
2.1.1 Deterministic Models 8
2.1.2 Stochastic Models 12
2.2 Nonlinear DAE Models 13
2.3 Linear DAE Models 24
2.3.1 Introduction 24
2.3.2 Regularity 25
2.3.3 A Canonical Form 27
2.3.4 Alternative Canonical Forms 31
2.3.5 State-Space Form 33
2.3.6 Sampling 36
2.4 Linear Time-Varying DAE Models 38
2.5 DAE Solvers 41
2.6 Linear Difference-Algebraic Equations 43
2.6.1 Regularity 44
2.6.2 A Canonical Form 44
2.6.3 State-Space Form 45
2.7 Stochastic Models 46
2.7.1 Stochastic Processes 47
2.7.2 Continuous-Time Linear Stochastic Models 48
2.7.3 Discrete-Time Linear Stochastic Models 50

2.7.4 Nonlinear Stochastic Models 53
2.8 Conclusions 53

3 System Identification 55
3.1 Prediction Error Methods 55
3.2 The Maximum Likelihood Method 57
3.3 Frequency Domain Identification 57
3.4 Identifiability 58
3.5 Observability 61

I Nonlinear DAE Models 63

4 Well-Posedness of Nonlinear Estimation Problems 65
4.1 Introduction 65
4.2 Literature Overview 67
4.3 Background and Motivation 67
4.4 Main Results 70
4.5 Particle Filtering 73
4.6 Implementation Issues 77
4.7 Example: Dymola Assisted Modeling and Particle Filtering 78
4.8 Parameter Estimation 82
4.9 Conclusions 85

5 Identifiability and Observability for DAEs Based on Kunkel and Mehrmann 87
5.1 Introduction 87
5.2 Identifiability 88
5.3 Observability Tests Based on Kunkel and Mehrmann 89
5.4 Identifiability Tests based on Kunkel and Mehrmann 90
5.5 Application to State-Space Models 93
5.6 Other Insights Using Kunkel's and Mehrmann's Theory 99
5.6.1 Observability Indices 99
5.6.2 Zero Dynamics 101
5.7 Conclusions 104

6 Identifiability Tests Using Differential Algebra for Component-Based Models 105
6.1 Introduction 105
6.2 Main Results 108
6.2.1 Global Identifiability 109
6.2.2 Local Identifiability 110
6.3 Applying the Results 110
6.4 Examples 111
6.5 A Mechanics Model Library 113
6.6 Conclusions 117

7 Simulation-Based Tests for Identifiability **119**
7.1 Introduction . . . 119
7.2 Basic Setup . . . 120
7.3 Examining Identifiability . . . 121
7.3.1 Preprocessing . . . 122
7.3.2 Drawing Conclusions on Identifiability . . . 124
7.3.3 Identifiable Functions of Parameters . . . 126
7.4 Example . . . 127
7.5 Conclusions and Ideas For Extensions . . . 129
7.5.1 Initialization for Identification . . . 129
7.5.2 Non-Minimum Phase Systems . . . 129
7.5.3 Trajectory Generation . . . 129
7.5.4 Observability . . . 129

II Linear DAE Models **131**

8 Linear SDAE Models **133**
8.1 Introduction . . . 133
8.2 Noise Modeling . . . 134
8.2.1 Time Domain Derivation . . . 135
8.2.2 Frequency Domain Derivation . . . 139
8.3 Example . . . 141
8.4 Sampling with Noise Model . . . 144
8.5 Kalman Filtering . . . 146
8.6 Time-Varying Linear SDAE Models . . . 146
8.7 Difference-Algebraic Equations . . . 148
8.7.1 Noise Modeling . . . 148
8.7.2 Kalman Filtering . . . 150
8.8 Conclusions . . . 150

9 Well-Posedness of Parameter Estimation Problems **151**
9.1 Introduction . . . 151
9.2 Problem Formulation . . . 151
9.3 Main Result . . . 153
9.4 Measuring Signals with Infinite Variance . . . 154
9.5 The Log-Likelihood Function and the Maximum Likelihood Method . . . 154
9.6 Frequency Domain Identification . . . 157
9.7 Time-Varying Linear SDAE Models . . . 158
9.8 Difference-Algebraic Equations . . . 160
9.8.1 Time Domain Identification . . . 160
9.8.2 Frequency Domain Identification . . . 161
9.9 Conclusions . . . 162

10 Well-Posedness of State Estimation Problems **163**
10.1 Introduction . . . 163
10.2 Formulations without Continuous-Time White Noise . . . 163
10.3 Formulations with Continuous-Time White Noise . . . 166
10.4 Example . . . 169
10.5 Time-Varying Linear SDAE Models . . . 170
10.6 Conclusions . . . 172

11 Implementation Issues **173**
11.1 Introduction . . . 173
11.2 Generalized Eigenvalues . . . 174
11.3 Computation of the Canonical Forms . . . 175
11.4 Summary . . . 176
11.5 Application Example . . . 178
11.6 Difference-Algebraic Equations . . . 179
11.7 Conclusions . . . 180

12 Initialization of Parameter Estimates **181**
12.1 Introduction . . . 181
12.2 Transforming the Problem . . . 183
12.2.1 The Case of Invertible $E(\theta)$ . . . 184
12.2.2 The Case of Non-Invertible $E(\theta)$ . . . 185
12.3 Sum of Squares Optimization . . . 187
12.4 Difference-Algebraic Equations . . . 188
12.5 Conclusions . . . 189

13 Conclusions **191**

A Notation **195**

B Proof of Theorem 9.1 **197**

Bibliography **201**

Index **207**

1

Introduction

Modeling of physical systems is a fundamental problem within the engineering sciences. Examples of physical systems that can be modeled are the weather, a human cell and an electrical motor. Models of these systems can differ greatly in complexity. For example, a model of the weather could be everything from a statement like "if it is sunny today, it is probably sunny tomorrow too" to a complex mathematical model used by meteorologists. Although models differ greatly in complexity, they have in common that they can be used to make predictions. A model of the weather could be used to make weather forecasts, a model of a human cell could be used to predict how it will react to different drugs, and a model of an electrical motor could be used to predict the effect of applying a certain voltage. In this thesis we will discuss mathematical models, that is equations that describe the behavior of a system. Such models can be constructed in different ways. One method is to use well-known physical relations, such as Newton's and Kirchhoff's laws. We will call this *physical modeling*. Another way is to estimate a model using measurements from the system. This is called *system identification*. For the electrical motor, we could for example measure the applied voltage and the resulting angle on the axis of the motor and estimate a model from that. A third case, which is a combination of the two previous modeling methods, is when we have constructed a model using physical relations but do not know the values of certain parameters in the model. These parameters could then be estimated using measurements from the system even if we cannot measure them directly. We will refer to this as *gray-box identification*.

Traditionally, physical modeling has been performed by manually writing down the equations that describe the system. If gray-box identification is necessary, the equations must be transformed manually into a suitable form. The manual modeling has today partly been replaced by tools that automate the physical modeling process. They both include tools for modeling systems within a certain domain, such as electrical systems, and general modeling tools that allow modeling of systems that contain components from different domains. An example of an object-oriented modeling language for multi-domain

modeling is Modelica (Fritzson, 2004; Tiller, 2001). These tools can greatly simplify the modeling task. In this thesis we will examine how gray-box identification in models generated by a modeling tool can be automated.

When a model has been constructed, it can be used to predict the future behavior of the modeled system and to estimate the values of variables that are not measured. We will thus also examine how models created using tools such as Modelica can be used for estimation and prediction.

1.1 Problem Formulation

In the most general setting, we would like to estimate parameters or unmeasured variables in a collection of equations that has been created by a modeling tool. These equations relate a vector of internal variables, $x(t)$, that vary with time, and their derivatives with respect to time, $\dot{x}(t)$, to inputs to the system $u(t)$. Here t denotes dependence on time. In the equations, there may be some unknown parameters θ that are to be estimated. An output $y(t)$ from the system is measured. The relationships can be described by the equations

$$F\big(\dot{x}(t), x(t), u(t), t, \theta\big) = 0 \tag{1.1a}$$

$$y(t) = h\big(x(t), \theta\big). \tag{1.1b}$$

This is called a differential-algebraic equation, or DAE (Brenan et al., 1996; Dai, 1989b; Kunkel and Mehrmann, 2001, 2006). DAEs are also known as descriptor systems, singular systems, and implicit systems. The discussion in this thesis concerns how the unknown constant parameters θ and unknown time-dependent variables $x(t)$ can be estimated using measurements of the input $u(t)$ and the output $y(t)$. Special attention will be given to how these problems can be approached by modeling disturbances acting on the system as stochastic processes.

Below we provide an example of a DAE. For this example it would be possible to transform the system into a form suitable for identification and estimation manually, but it would be much more convenient if the identification software could handle the DAE system directly. A form suitable for system is for example a state-space model, where (1.1a) takes the form $\dot{x}(t) = f\big(x(t), u(t), t, \theta\big)$.

Example 1.1: DAE model

Consider a cart which is driven forward by an electrical motor connected to one pair of the wheels. The parameters of a model of the system are the mass m, the radius of the wheels r, the torque constant of the motor k, the resistance R, the inductance L of the motor coil, and the coefficient b representing resistance caused by the air. The internal variables describing the system are the velocity of the cart $v(t)$, the acceleration of the cart $a(t)$, the force between the wheels and the ground $F(t)$, the torque from the motor $M(t)$, the angular velocity of the motor axis $\omega(t)$, some voltages in the motor, $u_L(t)$, $u_R(t)$, and $u_g(t)$, and the current $I(t)$. The input to the system is the voltage $u(t)$. If this system is modeled with a modeling tool such as in Figure 1.1, we get a collection of

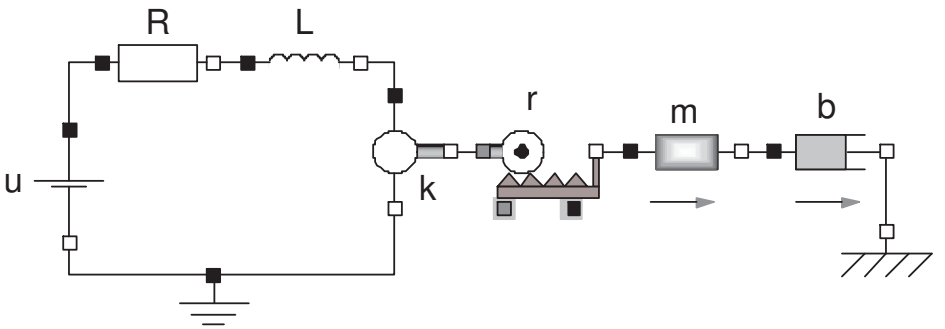


Figure 1.1: *A model produced using the modeling tool Modelica.*

equations describing the system, e.g.:

$$
\begin{aligned}
F(t) &= ma(t) & \frac{dv(t)}{dt} &= a(t) - bv^2(t) \\
F(t) &= rM(t) & r\omega(t) &= v(t) \\
M(t) &= kI(t) & u_g(t) &= k\omega(t) \\
u_R(t) &= RI(t) & u(t) &= u_L(t) + u_g(t) + u_R(t) \\
u_L(t) &= L\frac{dI(t)}{dt}
\end{aligned}
\tag{1.2}
$$

These equations could automatically be rewritten in the form (1.1). However, it would be tedious work to transform the equations into a form suitable for identification and estimation if we wanted the estimate one or more of the parameters or internal variables. How this process can be automated is one of the problems discussed in this thesis. Another problem which is discussed is how stochastic processes can be included in the equations to model disturbances acting on the system.

An important special case is when F and h in the DAE (1.1) are linear functions:

$$E(\theta)\dot{x}(t) = J(\theta)x(t) + K(\theta)u(t) \tag{1.3a}$$

$$y(t) = L(\theta)x(t) \tag{1.3b}$$

where $E(\theta)$, $J(\theta)$, $K(\theta)$, and $L(\theta)$ are matrices that contain unknown parameters θ that are to be estimated. The linear DAE (1.3) is also referred to as a linear descriptor system, a linear singular system and a linear implicit system. In the case with linear equations, analysis of system properties and other methods are better developed and easier to implement. The discrete-time counterpart of (1.3), where $\dot{x}(t)$ is replaced by $x(t+1)$, will also be discussed.

1.2 Outline

The purpose of this thesis is to describe how unknown parameters and time-dependent variables in DAE models can be estimated from measurement data. A background on the models and modeling techniques that are used in the thesis is provided in Chapter 2 and necessary background information on system identification is presented in Chapter 3. In the first part of the thesis, nonlinear DAE models are considered. Here noise modeling and estimation are discussed in Chapter 4 and the system properties identifiability and observability for DAE models are discussed in Chapter 5, 6, and 7. In the second part of the thesis, linear DAE models are considered. Noise modeling and estimation is discussed in Chapters 8, 9, and 10. Following that, implementation of estimation methods for linear DAE models is discussed in Chapter 11. Initialization of parameter estimates is discussed in Chapter 12.

The discussion in the thesis is concentrated on continuous-time DAE models, but it is described how most of the results for linear DAE models can be extended to the discrete-time case.

1.3 Contributions

The main contributions of the thesis are:

- The results on noise modeling in nonlinear DAE models and the discussion on how DAE models can be used in nonlinear particle filtering in Chapter 4.
- The application of the DAE theory by Kunkel and Mehrmann (2001) to examine observability and identifiability in Chapter 5.
- The results in Chapter 6 concerning how identifiability can be examined in two stages. If identifiability of the components of a model have been determined, identifiability of the complete model can be examined using a reduced number of equations.
- The results on how identifiability can be examined using DAE solvers in Chapter 7.
- The idea to redefine the input of a linear descriptor system to allow a state-space description for sampling (Chapter 2).
- The results on noise modeling for linear DAE models and how the noise model can be used for Kalman filtering and parameter estimation in Chapters 8, 9, and 10.
- The discussion on how the canonical forms for linear DAE models can be computed with the linear algebra package LAPACK (Chapter 11).
- The result that the parameter initialization problem under certain conditions can be transformed to the minimization of a biquadratic polynomial (Chapter 12).

The main results in Chapter 4 have been developed in cooperation with Johan Sjöberg and submitted as

M. Gerdin and J. Sjöberg. Nonlinear stochastic differential-algebraic equations with application to particle filtering. In *Proceedings of 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006. Accepted for publication.

The main results in Chapter 5 have been published in

M. Gerdin. Local identifiability and observability of nonlinear differential-algebraic equations. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006a,

the results in Chapter 6 have been published in

M. Gerdin and T. Glad. On identifiability of object-oriented models. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006,

and the results in Chapter 7 have been published in

M. Gerdin. Using DAE solvers to examine local identifiability for linear and nonlinear systems. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006b.

The main results in Chapter 8 have been developed in cooperation with Dr. Thomas Schön and Prof. Fredrik Gustafsson and have previously been published in

T. Schön, M. Gerdin, T. Glad, and F. Gustafsson. A modeling and filtering framework for linear differential-algebraic equations. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 892–897, Maui, Hawaii, USA, December 2003.

Part of the results in Chapter 9 and 11 have been submitted as

M. Gerdin, T. B. Schön, T. Glad, F. Gustafsson, and L. Ljung. On parameter and state estimation for linear differential-algebraic equations. *Automatica*, 2006. To appear,

and part of the results in Chapter 9 have been published in

M. Gerdin, T. Glad, and L. Ljung. Parameter estimation in linear differential-algebraic equations. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1530–1535, Rotterdam, the Netherlands, August 2003.

The results in Chapter 11 have also been published as

M. Gerdin. Computation of a canonical form for linear differential-algebraic equations. In *Proceedings of Reglermöte 2004*, Göteborg, Sweden, May 2004.

The results in Chapter 10 have been published as

M. Gerdin, T. Glad, and L. Ljung. Well-posedness of filtering problems for stochastic linear DAE models. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*, pages 350–355, Seville, Spain, December 2005.

2

Modeling

In this chapter, we will introduce the models and modeling techniques that are discussed in the thesis. We will also discuss general theory about these models and modeling techniques.

2.1 Introduction: Component-Based Modeling

By modeling, we mean the production of equations that can predict the behavior (or some part of the behavior) of a system. One way to model a physical system is to write down all equations describing the physics of a system. For larger systems this is of course a cumbersome approach. However, if parts of the system have been modeled previously, the work required can be reduced by reusing these parts. This leads to *component-based modeling*.

Component-based modeling is based on the need to combine models of different parts of a system to form a model of the complete system. The idea is that models of common parts, or components, are created once and for all, and stored in model libraries. They can then be reused when modeling larger systems. The components can typically be tuned for the application at hand by changing the value of parameters.

Consider for example electrical circuits. Typically, the equations describing components like resistors, capacitors, inductors, and voltage sources are stored in model libraries. The parameters of these components are the resistance, capacitance, inductance, and voltage respectively. When modeling a complete electrical circuit, the included components (including parameter values) and their interconnections are specified.

The modeling process can be simplified further by using modeling software with a graphical user interface where components can be selected and connected graphically. This makes it possible for a user to build complex models without having to deal with any equations. An example of a graphical model of an electrical circuit is shown in Figure 2.1.

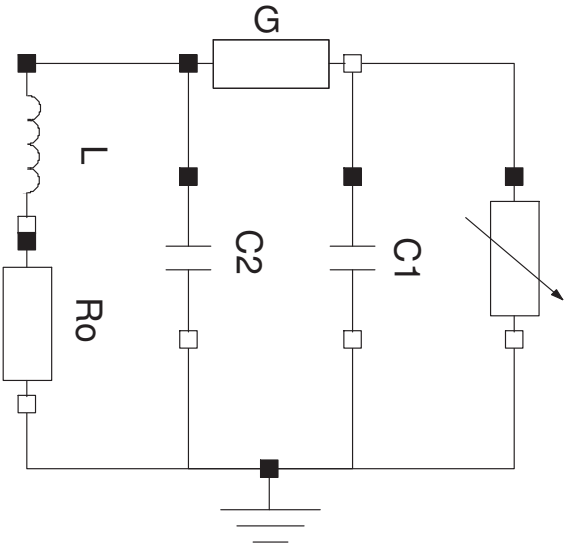


Figure 2.1: *A component-based model consisting of one linear resistor, one inductor, two capacitors, one conductor, one nonlinear resistor, and a ground point.*

The most wide-spread language for component-based modeling is Modelica (Fritzson, 2004; Tiller, 2001). Modelica is a *language* for component-based modeling. This is defined as a programming language specifically designed for component-based modeling. It includes constructs for example for defining equations and components (which can be connected with other components to create larger models) and possibilities to define graphical representations for the components. To use Modelica for modeling and simulation, an implementation of the language is necessary. Two implementations are Dymola and OpenModelica. The commercial product Dymola (Mattsson et al., 1998) is the most complete implementation available at the time of writing. OpenModelica (Fritzson, 2004, Chapter 19) is a free implementation with source code available. Modelica is often referred to as a language for *object-oriented* modeling. This concept includes the principles of component-based modeling, and also programming specific concepts such as classes and inheritance.

The main interest in this thesis is to examine how unknown parameters and internal variables in component-based models can be estimated. To do this, we will need to examine the structure of the equations that component-based modeling results in. This will be done in the following sections. We will also discuss how stochastic processes can be used to model disturbances.

2.1.1 Deterministic Models

By *deterministic models*, we mean models that do not include any stochastic variables or processes. Models that include stochastic processes or variables will be called *stochastic models* and are discussed in Section 2.1.2.

As discussed above, a component-based model consists of a number of components that each has a number of equations associated with it, and also equations describing how the components are connected. Each component i is described by equations f_i and the connections are described by equations g. To describe the behavior of the components, *internal variables* may be necessary. The internal variables of component i will be denoted l_i. The internal variables l_i are not involved in connections with other components. To describe connections with other components *external variables* w_i are used. These variables are used when describing how the components interact. The differentiation operator with respect to time is denoted p, $px(t) = \frac{d}{dt}x(t)$. The components may also contain constant unknown parameters θ_i. The collected parameters,

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_m \end{pmatrix} \tag{2.1}$$

are assumed to lie in a set $D_{\mathcal{M}} \subseteq \mathbb{R}^{n_\theta}$, $\theta \in D_{\mathcal{M}}$. Furthermore, known external stimuli on the system are denoted $u(t)$ and measured outputs are denoted $y(t)$.

Example 2.1: Component-based model

Consider the component-based model in Figure 2.1. It consists of seven components: two resistors, one conductor, two capacitors, one inductor, and one ground point. Let the inductor be component number 1. The external variables for this component are the potentials at the endpoints, $v_1(t)$ and $v_2(t)$, and the currents at the endpoints, $i_1(t)$ and $i_2(t)$. The internal variables are the voltage over the inductor $u_L(t)$ and the current through the inductor, $i_L(t)$. We thus have

$$w_1(t) = \begin{pmatrix} v_1(t) \\ v_2(t) \\ i_1(t) \\ i_2(t) \end{pmatrix} \text{ and } l_1(t) = \begin{pmatrix} u_L(t) \\ i_L(t) \end{pmatrix}. \tag{2.2}$$

The equations describing the inductor are

$$\underbrace{\begin{pmatrix} u_L(t) - \big(v_1(t) - v_2(t)\big) \\ i_1(t) - i_L(t) \\ i_2(t) - i_L(t) \\ u_L(t) - L \cdot pi_L(t) \end{pmatrix}}_{f_1} = 0. \tag{2.3}$$

Let the potential and current at the top endpoint of the resistor R_0 be denoted v_3 and i_3 respectively. The equations

$$\underbrace{\begin{pmatrix} v_2(t) - v_3(t) \\ i_2(t) - i_3(t) \end{pmatrix}}_{\text{part of } g} = 0 \tag{2.4}$$

would then be included in g to describe the connection between the inductor and the resistor R_0.

To summarize, the following parameters, variables, and equations are involved in a component-based model with m components.

- Internal variables l_i for each component $i = 1, \ldots, m$.
- External variables w_i for each component $i = 1, \ldots, m$. These are grouped as
$$w(t) = \begin{pmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_m(t) \end{pmatrix}. \tag{2.5}$$
- Unknown constant parameters θ_i for each component $i = 1, \ldots, m$. These are grouped as
$$\theta(t) = \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_m(t) \end{pmatrix}. \tag{2.6}$$
- Equations
$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \quad i = 1, \ldots, m, \tag{2.7}$$
describing each component. Note that this could be written without the differentiation operator p as
$$f_i\big(l_i(t), \dot{l}_i(t), \ddot{l}_i(t), \ldots, w_i(t), \dot{w}_i(t), \ddot{w}_i(t), \ldots, \theta_i\big) = 0 \quad i = 1, \ldots, m. \tag{2.8}$$
- Equations
$$g\big(u(t), w(t)\big) = 0 \tag{2.9}$$
describing the connections. Some of the $w_i(t)$ may be external stimuli, such as the voltage of a voltage source. This is specified through the known input function $u(t)$ in g.
- Variables $y(t)$ that are measured, are defined through the equation
$$y(t) = h\big(w(t), \theta\big) \tag{2.10}$$
where $y(t)$ is a measured signal. No l_i are included in this equation, since all signals that are visible to the outside of the component are included in w_i.

Collecting the equations gives the following model.

Model 1: Component-Based Model

$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \qquad i = 1, \ldots, m \tag{2.11a}$$
$$g\big(u(t), w(t)\big) = 0 \tag{2.11b}$$
$$y(t) = h\big(w(t), \theta\big) \tag{2.11c}$$

Note that f_i, g, l_i, w_i, θ_i, u, h, and y all may be vector valued.

Model 1 explicitly shows the structure of the equations that component-based modeling leads to. For example, the variables l_i are local for each component since only the w_i are involved in the connections. The equations $g = 0$ are typically simple equations like $a-b = 0$ or $a+b+c = 0$ and each equation normally only includes a small number of the w. This makes the system sparse, i.e., only a few variables are included in each equation. This special structure is utilized for example by the solvers for component-based models included in modeling environments like OpenModelica and Dymola. This is discussed further in Section 2.5. In Chapter 6 it is discussed how the structure can be used when examining identifiability of component-based models.

The equations can all be grouped together into one large system of equations to form a nonlinear differential-algebraic equation (DAE). DAE models are also known as descriptor systems, singular systems, and implicit systems.

Model 2: Nonlinear DAE

$$F\big(\dot{x}(t), x(t), \theta, u(t)\big) = 0 \tag{2.12a}$$

$$y(t) = h\big(x(t), \theta\big) \tag{2.12b}$$

In comparison with Model 1, the variables $l(t)$ and $w(t)$ have been grouped to form the vector of internal variables $x(t)$, the unknown parameters have been collected into one vector θ, and all equations have been included in the DAE $F = 0$. The equations are also written using only first order derivatives, $\dot{x}(t) = \frac{d}{dt}x(t)$. This is not a limitation, since higher derivatives can be specified by including additional variables. For example, $\ddot{x}(t)$ can be replaced by $\dot{z}(t)$ by including the equation $z(t) = \dot{x}(t)$. Nonlinear DAE models are further discussed in Section 2.2.

In the special case when all equations in Model 2 are linear in $x(t)$, $\dot{x}(t)$, and $u(t)$, we get a linear DAE.

Model 3: Linear DAE

$$E(\theta)\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \tag{2.13a}$$

$$y(t) = C(\theta)x(t) \tag{2.13b}$$

The reason to separate the linear DAE from its nonlinear counterpart, is that it is easier to analyze. For example, it is possible to reduce it to state-space form (an ordinary differential equation) explicitly under mild conditions. An analysis of linear DAEs is performed in Section 2.3.

An important special case of DAE models is when the derivative $\dot{x}(t)$ can be explicitly solved from the equations. The equations then form a *state-space model*,

$$\dot{x}(t) = f\big(x(t), \theta, u(t)\big) \tag{2.14a}$$

$$y(t) = h\big(x(t), \theta\big). \tag{2.14b}$$

If the equations are linear we have a linear state-space model,

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \tag{2.15a}$$
$$y(t) = C(\theta)x(t). \tag{2.15b}$$

Since state-space models are well examined in the literature, the techniques for state-space models will form the foundation for some of our discussion for DAE systems.

It can be noted that the idea to model a system by writing down all equations describing it, is related to the behavioral approach to modeling discussed by Polderman and Willems (1998).

2.1.2 Stochastic Models

In practice, mathematical models of physical systems cannot predict the exact behavior of the system, for example because of disturbances acting on the system. Disturbances can for example be wind acting on an aircraft or measurement noise in an electrical system. This motivates the introduction of stochastic models where such disturbances are modeled explicitly as stochastic processes. We will call disturbances that act on a system *noise* and we will make a distinction between *process noise* and *measurement noise*. Process noise is disturbances that affect the behavior of the system while measurement noise is disturbances that affect the measurements made on the system. Measurement noise is often modeled as additive on the measurement, and we will adopt this practice here. Denoting the process noise with $v_1(t)$ and the measurement noise with $v_2(t)$, a stochastic component-based model can be written as shown below.

Model 4: Stochastic Component-Based Model

$$f_i\big(l_i(t), w_i(t), \theta_i, p, v_{1,i}(t)\big) = 0 \qquad i = 1, \dots, m \tag{2.16a}$$
$$g\big(u(t), w(t)\big) = 0 \tag{2.16b}$$
$$y(t) = h\big(w(t), \theta\big) + v_2(t) \tag{2.16c}$$

The process noise has been divided into m parts, $v_{1,i}$ $i = 1, \dots, m$ to make explicit the fact that the different components typically are affected by different noise sources. As with deterministic models, the special structure of the component-based model may not always be possible to utilize. The equations and variables can then be grouped to form a nonlinear stochastic differential-algebraic equation (nonlinear SDAE).

Model 5: Nonlinear SDAE

$$F\big(\dot{x}(t), x(t), \theta, u(t), v_1(t)\big) = 0 \tag{2.17a}$$
$$y(t) = h\big(x(t), \theta\big) + v_2(t) \tag{2.17b}$$

If all equations are linear, we get a linear SDAE.

Model 6: Linear SDAE

$$E(\theta)\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + K(\theta)v_1(t) \tag{2.18a}$$
$$y(t) = C(\theta)x(t) + v_2(t) \tag{2.18b}$$

Special care must be taken when including stochastic processes in DAE models to make sure that the variables of interest are well-defined. For nonlinear SDAEs this will be discussed in Chapter 4, and for linear SDAEs it will be discussed in Chapters 8, 9, and 10.

The properties of stochastic models that will be used in the thesis are discussed in Section 2.7. As we will see, the stochastic state-space model is an important special case of stochastic DAE models. In the nonlinear case it can be written as

$$\dot{x}(t) = f\big(x(t), \theta, u(t)\big) + g\big(x(t), \theta, u(t)\big)v_1(t) \tag{2.19a}$$
$$y(t) = h\big(x(t), \theta\big) + v_2(t). \tag{2.19b}$$

We will limit the discussion to the case when the noise enters affinely according to the term $g\big(x(t), \theta, u(t)\big)v_1(t)$. We will discuss this further in Section 2.7. In the linear case a stochastic state-space model can be written as

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + K(\theta)v_1(t) \tag{2.20a}$$
$$y(t) = C(\theta)x(t) + v_2(t). \tag{2.20b}$$

These models are called stochastic differential equations (SDEs). As for the deterministic case, the theory for state-space models is well developed, and will be used when examining SDAE models.

2.2 Nonlinear DAE Models

As discussed above, a (deterministic) differential-algebraic equation (DAE) is a set of equations that can be written in the form

$$F\big(\dot{x}(t), x(t), u(t)\big) = 0. \tag{2.21}$$

DAE models are also known as descriptor systems, singular systems, and implicit systems. In this section we will review some of the theory available for such models. Since properties connected to unknown parameters and measured outputs are not discussed in this section, we omit these in the notation. Some general references on nonlinear DAEs are Brenan et al. (1996) that mainly discuss solution techniques and Kunkel and Mehrmann (2001) that discuss general properties of DAEs. The book by Kunkel and Mehrmann (2006) is also a good reference.

DAE models are in several ways more difficult to handle than state-space models. The difficulties center around the fact that it generally is not possible to solve (2.21) for $\dot{x}(t)$. If this was possible, the DAE could be written as a state-space system,

$$\dot{x}(t) = f\big(x(t), u(t)\big) \tag{2.22}$$

so that methods for state-space models could be used. To make it possible to transform a DAE into a state-space system (or similar description) it is usually necessary to differentiate the equations several times with respect to time. The following example illustrates this.

Example 2.2: Solving for $\dot{x}(t)$

Consider the DAE

$$\underbrace{\begin{pmatrix} \dot{x}_1(t) + x_1(t) \\ x_2(t) - x_1^2(t) \end{pmatrix}}_{F} = 0. \tag{2.23}$$

It is not possible to directly solve for

$$\dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} \tag{2.24}$$

as a function of $x(t)$. However, if the second equation of (2.23) is differentiated with respect to time we get

$$\begin{pmatrix} \dot{x}_1(t) + x_1(t) \\ \dot{x}_2(t) - 2x_1(t)\dot{x}_1(t) \end{pmatrix} = 0. \tag{2.25}$$

which can be solved for $\dot{x}(t)$ to give

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} -x_1(t) \\ 2x_1(t)\dot{x}_1(t) \end{pmatrix} = \begin{pmatrix} -x_1(t) \\ -2x_1^2(t) \end{pmatrix}. \tag{2.26}$$

The number of times a DAE must be differentiated in order to solve for $\dot{x}(t)$ is called the *differential index* or just the *index* of the system. The following definition is by Brenan et al. (1996).

Definition 2.1 (Differential index). The *differential index* of a DAE,

$$F\big(\dot{x}(t), x(t), u(t)\big) = 0, \tag{2.27}$$

is the smallest ν such that the system of equations

$$\begin{aligned} F\big(\dot{x}(t), x(t), u(t)\big) &= 0 \\ \frac{d}{dt} F\big(\dot{x}(t), x(t), u(t)\big) &= 0 \\ &\vdots \\ \frac{d^\nu}{dt^\nu} F\big(\dot{x}(t), x(t), u(t)\big) &= 0 \end{aligned} \tag{2.28}$$

uniquely determines the variable $\dot{x}(t)$ as a function of $x(t)$, $u(t)$, and derivatives of $u(t)$.

There are also other index concepts defined in the literature, but the differential index is the most common one. When the term *index* is used without specification, this typically refers to the differential index.

It is possible to form DAE systems that do not have a solution. Consider for example the following DAE.

Example 2.3: DAE without solution

The DAE

$$\dot{x}_1(t) + x_2(t) = 0 \tag{2.29a}$$
$$x_3(t) - t^2 = 0 \tag{2.29b}$$
$$x_3(t) + 5 = 0 \tag{2.29c}$$

does not have any solutions.

We will call those DAE systems where the corresponding initial value problem has at least one solution solvable DAEs.

Definition 2.2 (Solvable DAE). A DAE is called *solvable* if the corresponding initial value problem has at least one solution.

However, even if a DAE is solvable, it will in general not have solutions for all initial conditions $x(0)$. Consider the following example.

Example 2.4: Consistent initial conditions

The DAE

$$\dot{x}_1(t) + x_1(t) = 0 \tag{2.30a}$$
$$x_1(t) - x_2^3(t) = 0 \tag{2.30b}$$

has solutions only for initial conditions satisfying $x_1(0) - x_2^3(0) = 0$.

Those initial conditions for which the initial value problem has a solution will be called consistent initial conditions.

Definition 2.3 (Consistent initial conditions). The *consistent initial conditions* of a DAE are the initial conditions such that the corresponding initial value problem has at least one solution.

Kunkel and Mehrmann's Analysis

One method to examine DAE systems in presented by Kunkel and Mehrmann (2001), and these results are summarized in this section. For proofs and a complete discussion, the reader is referred to Kunkel and Mehrmann (2001). A slightly modified presentation of the results is included in the book by Kunkel and Mehrmann (2006). The results are based on rank tests and the implicit function theorem, and are therefore only valid locally.

Their most important result for the purpose of this thesis is that, provided a number of conditions are satisfied, it is possible to view the DAE as a combination of ordinary differential equations that determine one part of the variables (denoted x_1) and algebraic equations determining another part of the variables (denoted x_3). If some variables are not determined by the equations, then these are denoted x_2. For a control engineer, the x_2 variables can often be seen as inputs to the system or external stimuli. This division of the variables is illustrated by the following example.

Example 2.5: Separating the internal variables

Consider the DAE

$$\underbrace{\begin{pmatrix} \dot{x}_1 - x_3^2 \\ x_3 - x_2^3 - t \end{pmatrix}}_{F} = 0. \tag{2.31}$$

This DAE can be written as

$$\dot{x}_1 = x_3^2 \tag{2.32}$$

$$x_3 = x_2^3 + t \tag{2.33}$$

where we see that x_1 is determined by an ordinary differential equation, x_3 is determined by a static relationship, and x_2 can be seen as an external stimulus. Note that it would be possible to exchange the roles of x_2 and x_3 in this example. Also note that x_1, x_2, and x_3 are scalars in this example, but generally they are vectors.

This example will be used throughout this section to illustrate the concepts discussed.

To simplify notation, we let the DAE depend directly on the time variable t instead of an input signal $u(t)$,

$$F\big(\dot{x}(t), x(t), t\big) = 0 \tag{2.34}$$

where

$$F \in \mathbb{R}^m \tag{2.35a}$$

$$x \in \mathbb{R}^n \tag{2.35b}$$

$$t \in \mathbb{I} \tag{2.35c}$$

where $\mathbb{I} \subseteq \mathbb{R}$ is a compact interval. Kunkel and Mehrmann's analysis is based on successive differentiations of the DAE. Therefore define a nonlinear *derivative array*

$$F_l(t, x, \dot{x}, \ldots, x^{l+1}) = 0 \tag{2.36}$$

which stacks the original equations and all their derivatives up to level l:

$$F_l(t, x, \dot{x}, \ldots, x^{l+1}) = \begin{pmatrix} F(\dot{x}, x, t) \\ \frac{d}{dt} F(\dot{x}, x, t) \\ \vdots \\ \frac{d^l}{dt^l} F(\dot{x}, x, t) \end{pmatrix} \tag{2.37}$$

Example 2.6: Derivative array

For the DAE in Example 2.5 we have

$$F_0 = F = \begin{pmatrix} \dot{x}_1 - x_3^2 \\ x_3 - x_2^3 - t \end{pmatrix} \tag{2.38}$$

and

$$F_1 = \begin{pmatrix} F \\ \frac{d}{dt}F \end{pmatrix} = \begin{pmatrix} \dot{x}_1 - x_3^2 \\ x_3 - x_2^3 - t \\ \ddot{x}_1 - 2x_3\dot{x}_3 \\ \dot{x}_3 - 3x_2^2\dot{x}_2 - 1 \end{pmatrix}. \tag{2.39}$$

Partial derivatives of F_l with respect to selected variables p from $(t, x, \dot{x}, \ldots, x^{(l+1)})$ are denoted by $F_{l;p}$, e.g.,

$$F_{l;\dot{x},\ldots,x^{(l+1)}} = \begin{pmatrix} \frac{\partial}{\partial \dot{x}}F_l & \frac{\partial}{\partial \ddot{x}}F_l & \cdots & \frac{\partial}{\partial x^{(l+1)}}F_l \end{pmatrix}. \tag{2.40}$$

A corresponding notation is used for partial derivatives of other functions.

Example 2.7: Notation for partial derivatives

For the DAE in example in Example 2.5 we have

$$F_{0;x,\dot{x}} = \begin{pmatrix} 0 & 0 & -2x_3 & 1 & 0 & 0 \\ 0 & -3x_2^2 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{2.41}$$

$$F_{0;x} = \begin{pmatrix} 0 & 0 & -2x_3 \\ 0 & -3x_2^2 & 1 \end{pmatrix}, \tag{2.42}$$

and

$$F_{0;\dot{x}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.43}$$

The solution of the derivative array F_μ for some integer μ is denoted

$$\mathbb{L}_\mu = \{ z_\mu \in \mathbb{I} \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \,|\, F_\mu(z_\mu) = 0 \}. \tag{2.44}$$

Example 2.8: $\mathbb{L}_\mu$ — solution of the derivative array

For the DAE in example in Example 2.5 we have, with $\mu = 0$,

$$\mathbb{L}_0 = \left\{ z_0 \in \mathbb{I} \times \mathbb{R}^3 \times \mathbb{R}^3 \,\middle|\, \begin{pmatrix} z_{0,5} - z_{0,4}^2 \\ z_{0,4} - z_{0,3}^3 - z_{0,1} \end{pmatrix} = 0 \right\} \tag{2.45}$$

where $z_{0,i}$ represents the i:th element of the vector z_0. We thus have $z_{0,1}$ representing the time t, $z_{0,2}$ representing x_1, $z_{0,3}$ representing x_2, $z_{0,4}$ representing x_3, $z_{0,5}$ representing $\dot{x}_1$ and so on. The set $\mathbb{L}_0$ is shown in Figure 2.2.

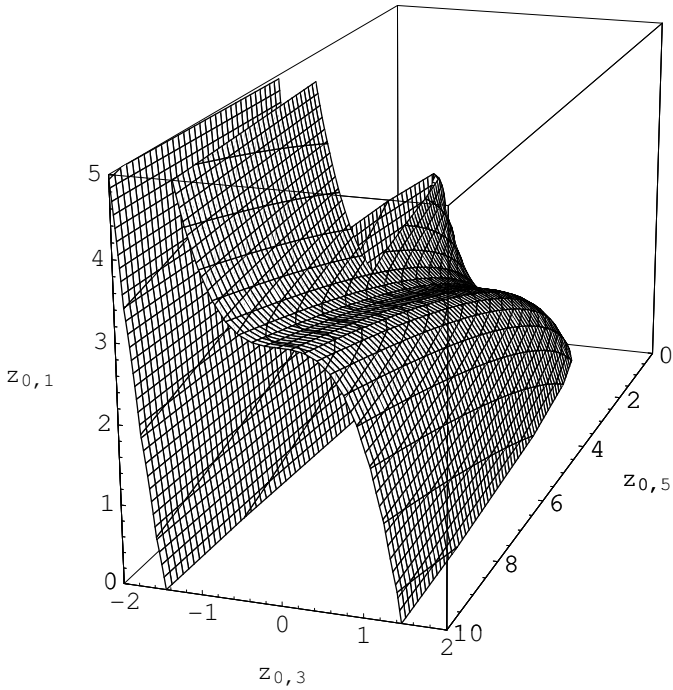


Figure 2.2: *The set $\mathbb{L}_0$ in Example 2.8. For the variables in z_0 that are not shown, we have that $z_{0,4} = z_{0,3}^3 + z_{0,1}$ and that $z_{0,2}$, $z_{0,6}$, and $z_{0,7}$ can take arbitrary values in $\mathbb{R}$.*

To present the main results, the corank of a matrix must be defined (Kunkel and Mehrmann, 2005, page 374).

Definition 2.4 (Corank). The *corank* of a matrix is the rank deficiency with respect to rows. The convention that $\operatorname{corank} F_{-1;x} = 0$ is used.

For example, if a matrix has 5 rows and rank 3, the corank is 2. The following property, Hypothesis 1 by Kunkel and Mehrmann (2001), which describes the basic requirements on DAE models handled by the theory can now be formulated.

Property 2.1

Consider the general nonlinear DAE (2.34). There exist integers μ, r, a, d, and v such that the following conditions hold:

1. The set $\mathbb{L}_\mu \subseteq \mathbb{R}^{(\mu+2)n+1}$ forms a manifold of dimension $(\mu+2)n+1-r$.
2. We have
$$\operatorname{rank} F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}} = r \tag{2.46}$$
on $\mathbb{L}_\mu$.

3. We have
$$\operatorname{corank} F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}} - \operatorname{corank} F_{\mu-1;x,\dot{x},\ldots,x^{(\mu)}} = v \tag{2.47}$$
on $\mathbb{L}_\mu$.

4. We have
$$\operatorname{rank} F_{\mu;\dot{x},\ldots,x^{(\mu+1)}} = r - a \tag{2.48}$$
on $\mathbb{L}_\mu$ such that there are smooth full rank matrix functions Z_2 and T_2 defined on $\mathbb{L}_\mu$ of size $\big((\mu+1)m, a\big)$ and $(n, n-a)$, respectively, satisfying
$$Z_2^T F_{\mu;\dot{x},\ldots,x^{(\mu+1)}} = 0 \tag{2.49a}$$
$$\operatorname{rank} Z_2^T F_{\mu;x} = a \tag{2.49b}$$
$$Z_2^T F_{\mu;x} T_2 = 0 \tag{2.49c}$$
on $\mathbb{L}_\mu$.

5. We have
$$\operatorname{rank} F_{\dot{x}} T_2 = d = m - a - v \tag{2.50}$$
on $\mathbb{L}_\mu$.

One of the more restrictive assumptions of Property 2.1 is that all ranks are constant. This for example rules out models where the number of states changes over time. Apart from that, the assumptions of Property 2.1 are not very restrictive and are satisfied for many physical models.

The property makes it possible to define a new index concept, the *strangeness index* (Kunkel and Mehrmann, 2001, 2006).

Definition 2.5 (Strangeness index). The *strangeness index* of the DAE system (2.34) is the smallest μ such that Property 2.1 is satisfied.

Example 2.9: Verifying Property 2.1

In this example, we show that the system in Example 2.5 fulfills Property 2.1 with $\mu = 0$, and thus has strangeness index 0. Note that the dimension of x is $n = 3$ and the dimension of F is $m = 2$. Take $\mu = 0$. The parts of Property 2.1 can now be verified as follows.

1. From (2.45) and Figure 2.2 we get that $\mathbb{L}_0$ forms a manifold of dimension 5. We must thus have
$$r = (\mu+2)n + 1 - 5 = 2. \tag{2.51}$$

2. From (2.41) we get that
$$\operatorname{rank} F_{0;x,\dot{x}} = 2 \tag{2.52}$$
which is consistent with $r = 2$.

3. $F_{0;x,\dot{x}}$ has full row rank, so $v = 0$.

4. From (2.43) we get that
$$\operatorname{rank} F_{0,\dot{x}} = 1 \tag{2.53}$$
which gives
$$a = r - 1 = 1. \tag{2.54}$$
The matrices Z_2 and T_2 can be taken as
$$Z_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.55}$$
and
$$T_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 3x_2^2 \end{pmatrix}. \tag{2.56}$$

5. We have
$$\operatorname{rank} F_{\dot{x}} T_2 = \operatorname{rank} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 3x_2^2 \end{pmatrix} = \operatorname{rank} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = 1. \tag{2.57}$$
This is consistent with
$$m - a - v = 2 - 1 - 0 = 1, \tag{2.58}$$
so we have a well-defined $d = 1$.

For DAE models that satisfy the property it is, as mentioned in the beginning of the section, possible to divide the equations into one part that (locally) forms an ordinary differential equation for one part of the variables (denoted x_1), and one part that locally forms static equations that determine another part of the variables (denoted x_3). If any variables still are undetermined, then they can be chosen freely. These variables are denoted x_2. Note that no variable transformation is necessary, only a reordering of the variables needs to be done. This means that we can write
$$x(t) = Q \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix}, \; Q \text{ permutation matrix.} \tag{2.59}$$
This transformation is performed by first defining the $m \times d$ matrix Z_1 (which can be chosen constant) such that
$$\operatorname{rank} Z_1^T F_{\dot{x}} T_2 = d \tag{2.60}$$
and then forming
$$\hat{F}_1 = Z_1^T F \tag{2.61a}$$
$$\hat{F}_2 = Z_2^T F_\mu. \tag{2.61b}$$

This gives the equations

$$\hat{F}_1(t, x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3) = 0 \tag{2.62a}$$

$$\hat{F}_2(t, x_1, x_2, x_3) = 0. \tag{2.62b}$$

Example 2.10: Computing $\hat{F}_1$ and $\hat{F}_2$

For the DAE in example 2.5 we can take

$$Z_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.63}$$

and we thus get

$$\hat{F}_1 = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{Z_1^T} \underbrace{\begin{pmatrix} \dot{x}_1 - x_3^2 \\ x_3 - x_2^3 - t \end{pmatrix}}_{F} = \dot{x}_1 - x_3^2 \tag{2.64a}$$

$$\hat{F}_2 = \underbrace{\begin{pmatrix} 0 & 1 \end{pmatrix}}_{Z_2^T} \underbrace{\begin{pmatrix} \dot{x}_1 - x_3^2 \\ x_3 - x_2^3 - t \end{pmatrix}}_{F_0} = x_3 - x_2^3 - t. \tag{2.64b}$$

$\hat{F}_2 = 0$ can (locally) be solved for x_3 to give the equations

$$\hat{F}_1(t, x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3) = 0 \tag{2.65a}$$

$$x_3 = \mathcal{R}(t, x_1, x_2). \tag{2.65b}$$

After using (2.65b) to eliminate x_3 and $\dot{x}_3$ in (2.65a), (2.65a) can be locally solved for $\dot{x}_1$ to give

$$\dot{x}_1 = \mathcal{L}(t, x_1, x_2, \dot{x}_2) \tag{2.66a}$$

$$x_3 = \mathcal{R}(t, x_1, x_2). \tag{2.66b}$$

Example 2.11: Separating the internal variables

For the DAE in Example 2.5 we can solve $\hat{F}_1 = 0$ and $\hat{F}_2 = 0$ for $\dot{x}_1$ and x_3:

$$\dot{x}_1 = x_3^2 \tag{2.67a}$$

$$x_3 = x_2^3 + t. \tag{2.67b}$$

Eliminating x_3 in (2.67a) using (2.67b) gives

$$\dot{x}_1 = \underbrace{(x_2^3 + t)^2}_{\mathcal{L}} \tag{2.68a}$$

$$x_3 = \underbrace{x_2^3 + t}_{\mathcal{R}}. \tag{2.68b}$$

The above discussion can be summarized by the following theorem, which is a version of Theorem 3 by Kunkel and Mehrmann (2001).

Theorem 2.1
Let F in (2.34) *satisfy Property 2.1 with μ, a, d, v. Then every solution of* (2.34) *solves a reduced problem,*

$$\dot{x}_1 = \mathcal{L}(t, x_1, x_2, \dot{x}_2) \tag{2.69a}$$
$$x_3 = \mathcal{R}(t, x_1, x_2) \tag{2.69b}$$

consisting of d differential and a algebraic equations. The elements of $x_1 \in \mathbb{R}^d$, $x_2 \in \mathbb{R}^{n-a-d}$, and $x_3 \in \mathbb{R}^a$ together make up the elements of x.

Proof: See Theorem 3 in Kunkel and Mehrmann (2001). □

Note that it is typically not possible to solve for $\dot{x}_1$ and x_3 explicitly (the existence of the transition is proved using the implicit function theorem). Instead it is usually necessary to work with $\hat{F}_1$ and $\hat{F}_2$ and solve for $\dot{x}_1$ and x_3 numerically. However, it is possible to solve explicitly for $\dot{x}_3$. This can be seen by differentiating (2.62b) with respect to time,

$$\frac{d}{dt}\hat{F}_2 = \hat{F}_{2;t} + \hat{F}_{2;x_1}\dot{x}_1 + \hat{F}_{2;x_2}\dot{x}_2 + \hat{F}_{2;x_3}\dot{x}_3. \tag{2.70}$$

Since $\hat{F}_2$ can be solved locally for x_3, $\hat{F}_{2;x_3}$ is non-singular. This means that $\dot{x}_3$ can be written as

$$\dot{x}_3 = -\hat{F}_{2;x_3}^{-1}\left(\hat{F}_{2;t} + \hat{F}_{2;x_1}\dot{x}_1 + \hat{F}_{2;x_2}\dot{x}_2\right) \tag{2.71}$$

where $\hat{F}_{2;x_3}^{-1}$ is the inverse of the matrix $\hat{F}_{2;x_3}$. We can thus expect to work with equations like

$$\tilde{F}_1(t, x_1, x_2, x_3, \dot{x}_1, \dot{x}_2) = 0 \tag{2.72a}$$
$$\hat{F}_2(t, x_1, x_2, x_3) = 0 \tag{2.72b}$$

where $\tilde{F}_1$ is $\hat{F}_1$ with $\dot{x}_3$ eliminated using (2.71).

The theorem above states that every solution of the DAE also solves the reduced system. To show that the solutions of the reduced systems solve the original DAE, additional requirements must be fulfilled as stated by the following theorem. In this theorem, Property 2.1 must be satisfied for two successive values of μ with the other constants in the property unchanged.

Theorem 2.2
Let F in (2.34) *be sufficiently smooth and satisfy Property 2.1 with μ, a, d, v and with $\mu + 1$ (replacing μ), a, d, v. Let the initial condition $z^0_{\mu+1} \in \mathbb{L}_{\mu+1}$ be given and let the parameterization of the solution of $F_{\mu+1}$ include $\dot{x}_2$. Then, for every function $x_2 \in C^1 \cdot (\mathbb{I}, \mathbb{R}^{n-a-d})$ that satisfies $z^0_{\mu+1}$, the reduced system*

$$\dot{x}_1 = \mathcal{L}(t, x_1, x_2, \dot{x}_2) \tag{2.73a}$$
$$x_3 = \mathcal{R}(t, x_1, x_2) \tag{2.73b}$$

has unique solutions x_1 and x_3 where x_1 satisfies the initial condition $z^0_{\mu+1}$. Moreover, these together locally solve the original problem.

Proof: See Theorem 4 in Kunkel and Mehrmann (2001). □

The term "locally solves" in this theorem refers to the fact that solutions to (2.73) only represents one set of solutions to the original DAE. There could possibly be solutions with other properties. For example, the equation $x = \sqrt{y}$ "locally solves" $x^2 - y = 0$ for $y > 0$.

It is possible to select the initial value $x_1(0)$ freely in a neighborhood of each possible value as noted by the following proposition.

Proposition 2.1
Let F in (2.34) *satisfy the conditions of Theorem 2.2. Let x_{10} be the part of $z^0_{\mu+1} \in \mathbb{L}_{\mu+1}$ belonging to x_1. If $\tilde{x}_{10}$ is sufficiently close to x_{10}, it is part of a $\tilde{z}^0_{\mu+1} \in \mathbb{L}_{\mu+1}$ close to $z^0_{\mu+1}$ and Theorem 2.2 can be applied with $z^0_{\mu+1}$ replaced by $\tilde{z}^0_{\mu+1}$. The same holds for x_2.*

Proof: It follows from the proof of Theorem 4 in Kunkel and Mehrmann (2001) that $\mathbb{L}_{\mu+1}$ locally can be parameterized by t, x_1, x_2, and p where p is chosen from $\dot{x} \dots x^{\mu+2}$. $\tilde{x}_{10}$ can thus be chosen freely if it is sufficiently close to x_{10}. The same holds for x_2. □

If there are no free parameters x_2, Theorem 2.2 simplifies to the following corollary.

Corollary 2.1
Let F in (2.34) *be sufficiently smooth and satisfy Property 2.1 with μ, a, d, v and with $\mu + 1$ (replacing μ), a, d, v, and assume that $a + d = n$. Fore every $z^0_{\mu+1} \in \mathbb{L}_{\mu+1}$ the reduced problem*

$$\dot{x}_1 = \mathcal{L}(t, x_1) \tag{2.74a}$$

$$x_3 = \mathcal{R}(t, x_1) \tag{2.74b}$$

has a unique solution satisfying the initial value given by $z^0_{\mu+1}$. Moreover, this solution locally solves the original problem.

Proof: See Corollary 5 in Kunkel and Mehrmann (2001). □

The forms discussed here will be important tools when examining noise modeling, identifiability, and observability for nonlinear DAE models. The transformation is typically not unique, for example there may be different possible choices of state variables x_1. It is also common that the DAE is well determined so that x_2 has size zero, as for example in Corollary 2.1. This is defined as regularity.

Definition 2.6 (Regularity). The DAE

$$F\big(\dot{x}(t), x(t), u(t)\big) = 0 \tag{2.75}$$

is called *regular* if it satisfies Property 2.1 and $n - a - d = 0$ or, equivalently, the size of x_2 is equal to zero.

When using the method discussed in this section, it is usually necessary to successively increase μ until the property is true. The property could for example be verified by numeric rank tests at a certain value of $x(t)$, see further Remark 1 by Kunkel and Mehrmann (2001). The practical implementation of methods related to the property is also discussed in a more recent paper by Kunkel and Mehrmann (2004).

2.3 Linear DAE Models

In this section we will discuss some concepts concerning linear DAE systems that will be needed to motivate or develop the theory discussed in later chapters. The theory for linear DAE systems is presented separately since the linear structure allows a detailed analysis than for the nonlinear case. Linear DAEs are also known as linear descriptor systems, linear singular systems, and linear implicit systems.

2.3.1 Introduction

A linear DAE is a system of equations in the form

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.76a}$$

$$y(t) = Lx(t). \tag{2.76b}$$

In this description E and J are constant square matrices and K and L are constant matrices. Note that E may be a singular matrix. This makes it possible to include a purely algebraic equation in the description for example by letting a row of E be equal to zero. The vectors $u(t)$ and $y(t)$ are the input and the measured output respectively. Finally, the vector $x(t)$ contains the internal variables that describe the current state of the system.

It is also possible to form a discrete-time counterpart of the linear DAE (2.76).

$$Ex(t+1) = Jx(t) + Ku(t) \tag{2.77a}$$

$$y(t) = Lx(t) \tag{2.77b}$$

This model is called a system of difference-algebraic equations or a discrete-time descriptor system.

Two references on linear DAE systems are the book by Dai (1989b) and the survey by Lewis (1986). They discuss both general properties of linear DAE systems such as regularity and canonical forms, as well as controllability, observability, and different control and estimation strategies. They are both focused on the continuous-time case, but also treat discrete-time systems. Many references to earlier work are provided by both authors. Within the numerical analysis literature, Brenan et al. (1996) is worth mentioning. The main topic is the numerical solution of nonlinear DAEs, but linear DAE systems are also treated. One can also note that linear DAE systems are special cases of the *general linear constant differential equations* discussed by Rosenbrock (1970). Rosenbrock's analysis is mainly carried out in the frequency domain. Linear DAE systems are also special cases of the *general differential systems* discussed by Kailath (1980, Chapter 8). Linear DAE models are also discussed in the book by Kunkel and Mehrmann (2006).

The main topics of this section is to describe how the linear DAE system (2.76) can be transformed into different canonical forms (Section 2.3.3), and how it can be further transformed into a state-space system with a redefined input (Section 2.3.5). It is also discussed how a linear DAE system can be sampled by first transforming it to state-space form in Section 2.3.6. In Section 2.6 the results for the continuous-time case are extended to the discrete-time case.

Before proceeding into details of the canonical forms it may be worthwhile to note that (2.76) has the transfer function

$$G(s) = L(sE - J)^{-1}K. \tag{2.78}$$

A difference between $G(s)$ in (2.78) and the transfer function of a state-space system is that $G(s)$ in (2.78) may be non-proper (have higher degree in the numerator than in the denominator) in the general case. This can be realized from the following example:

$$\left(s \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{E} - \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{J} \right)^{-1} = - \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \tag{2.79}$$

It can be noted that the transfer function in (2.78) only is well-defined if $(sE - J)$ is non-singular. In Section 2.3.2 we will define the non-singularity of this matrix as regularity of the system (2.76). We will also see that regularity of a linear DAE system is equivalent to the existence of a unique solution.

2.3.2 Regularity

A basic assumption which is made throughout this thesis is that the inverse in (2.78) is well-defined, and therefore we formalize this with a definition.

Definition 2.7 (Regularity). The linear DAE system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.80a}$$

$$y(t) = Lx(t) \tag{2.80b}$$

is called *regular* if

$$\det(sE - J) \not\equiv 0, \tag{2.81}$$

that is the determinant is not zero for all s.

This definition is the same as the one used by Dai (1989b). The reason that regularity of a linear DAE system is a reasonable assumption, is that it is equivalent to the existence of a unique solution, as discussed by Dai (1989b, Chapter 1). To illustrate this, we examine the Laplace transformed version of (2.80a):

$$sE\mathcal{L}\left[x(t)\right] - Ex(0) = J\mathcal{L}\left[x(t)\right] + K\mathcal{L}\left[u(t)\right] \tag{2.82}$$

where $\mathcal{L}[\cdot]$ means the Laplace transform of the argument. Rearranging this, we get

$$(sE - J)\,\mathcal{L}\left[x(t)\right] = K\mathcal{L}\left[u(t)\right] + Ex(0) \tag{2.83}$$

If the system is regular, we get that $\mathcal{L}\left[x(t)\right]$ is uniquely determined by

$$\mathcal{L}\left[x(t)\right] = (sE - J)^{-1}\left(K\mathcal{L}\left[u(t)\right] + Ex(0)\right). \tag{2.84}$$

If, on the other hand, the system is not regular, there exists a vector $\alpha(s) \not\equiv 0$ such that

$$(sE - J)\alpha(s) \equiv 0. \tag{2.85}$$

We get that if the system is not regular and a solution of (2.83) is $\mathcal{L}\left[x(t)\right]$, then so is $\mathcal{L}\left[x(t)\right] + k\alpha(s)$ for any constant k. A solution is consequently not unique. It is also

obvious that a solution may not even exist if the system is not regular, for example if $(sE - J) \equiv 0$.

To draw conclusions about $x(t)$ from the existence of $\mathcal{L}[x(t)]$, we should examine if the inverse Laplace transform exists. We do not go into these technicalities here. However, in the next section we will see how a regular linear DAE system can be transformed into a form where the existence of a solution is obvious.

It is usually a reasonable assumption that a system has an input which uniquely determines the value of the internal variables for each consistent initial condition. With this motivation, it will be assumed throughout this thesis that the systems encountered are regular.

We conclude this section with a small example to illustrate the connection between solvability and regularity.

Example 2.12: Regularity

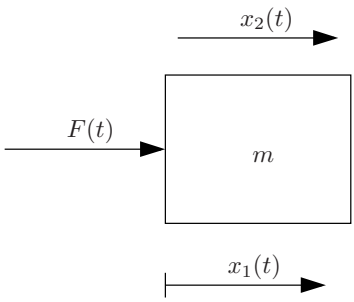


Figure 2.3: *A body affected by a force.*

Consider the body with mass m in Figure 2.3. The body has position $x_1(t)$ and velocity $x_2(t)$ and is affected by a force $F(t)$. The equations describing the system are

$$\dot{x}_1(t) = x_2(t) \tag{2.86a}$$

$$m\dot{x}_2(t) = F(t) \tag{2.86b}$$

which also can be written as

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix}}_{E} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{J} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{K} F(t) \tag{2.87}$$

which is a linear DAE system (without output equation). We get that

$$\det(sE - J) = ms^2 \tag{2.88}$$

and the system is regular if and only if $m \neq 0$. According to the discussion earlier this gives that there exists a unique solution if and only if $m \neq 0$. This is also obvious from the original equations (2.86). In this example we also see that regularity is a reasonable requirement on the system.

2.3.3 A Canonical Form

In this section we examine how a linear DAE system can be rewritten in a form which resembles a state-space system and explicitly shows how the solution of the DAE system can be obtained. This transformation will later play an important role in the development of the identification algorithms. Similar transformations have been considered earlier in the literature (see e.g., Dai, 1989b), but the proofs which are presented in this section have been constructed so that the indicated calculations can be computed by numerical software in a reliable manner. How the different steps of the proofs can be computed numerically is studied in detail in Chapter 11. It can be noted that the system must be regular for the transformation to exist, but as discussed in Section 2.3.2 regularity is equivalent to the existence of a unique solution.

The main result is presented in Theorem 2.3, but to derive this result we use a series of lemmas as described below. The first lemma describes how the system matrices E and J simultaneously can be written in triangular form with the zero diagonal elements of E sorted to the lower right block.

Lemma 2.1
Consider a system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.89a}$$

$$y(t) = Lx(t). \tag{2.89b}$$

If (2.89) *is regular, then there exist non-singular matrices P_1 and Q_1 such that*

$$P_1 E Q_1 = \begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} \text{ and } P_1 J Q_1 = \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix} \tag{2.90}$$

where E_1 is non-singular, E_3 is upper triangular with all diagonal elements zero and J_3 is non-singular and upper triangular.

Note that either the first or the second block row in (2.90) may be of size zero.

Proof: The Kronecker canonical form of a regular matrix pencil which is discussed by, e.g., Kailath (1980, Chapter 6) immediately shows that it is possible to perform the transformation (2.90). □

In the case when the matrix pencil is regular, the Kronecker canonical form is also called the Weierstrass canonical form. The Kronecker and Weierstrass canonical forms are also discussed by (Gantmacher, 1960, Chapter 12). The original works are by Weierstrass (1868) and Kronecker (1890).

Note that the full Kronecker form is not computed by the numerical software discussed in Chapter 11. The Kronecker form is here just a convenient way of showing that the transformation (2.90) is possible.

The next two lemmas describe how the internal variables of the system can be separated into two parts by making the system matrices block diagonal.

Lemma 2.2
Consider (2.90). *There exist matrices L and R such that*

$$\begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} = \begin{pmatrix} E_1 & 0 \\ 0 & E_3 \end{pmatrix} \tag{2.91}$$

and

$$\begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} = \begin{pmatrix} J_1 & 0 \\ 0 & J_3 \end{pmatrix}. \tag{2.92}$$

Proof: See Kågström (1994) and references therein for a proof of this lemma. □

Lemma 2.3
Consider a system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.93a}$$
$$y(t) = Lx(t). \tag{2.93b}$$

If (2.93) *is regular, there exist non-singular matrices P and Q such that the transformation*

$$PEQQ^{-1}\dot{x}(t) = PJQQ^{-1}x(t) + PKu(t) \tag{2.94}$$

gives the system

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\dot{x}(t) = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} Q^{-1}x(t) + \begin{pmatrix} B \\ D \end{pmatrix} u(t) \tag{2.95}$$

where N is a nilpotent matrix.

Proof: Let P_1 and Q_1 be the matrices in Lemma 2.1 and define

$$P_2 = \begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \tag{2.96a}$$
$$Q_2 = \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \tag{2.96b}$$
$$P_3 = \begin{pmatrix} E_1^{-1} & 0 \\ 0 & J_3^{-1} \end{pmatrix} \tag{2.96c}$$

where L and R are from Lemma 2.2. Also let

$$P = P_3P_2P_1 \tag{2.97a}$$
$$Q = Q_1Q_2. \tag{2.97b}$$

Then

$$PEQ = \begin{pmatrix} I & 0 \\ 0 & J_3^{-1}E_3 \end{pmatrix} \tag{2.98}$$

and

$$PJQ = \begin{pmatrix} E_1^{-1}J_1 & 0 \\ 0 & I \end{pmatrix} \tag{2.99}$$

Here $N = J_3^{-1}E_3$ is nilpotent since E_3 is upper triangular with zero diagonal elements and J_3^{-1} is upper triangular. J_3^{-1} is upper triangular since J_3 is. Defining $A = E_1^{-1}J_1$ finally gives us the desired form (2.95). □

We are now ready to present the main result in this section, which shows how the solution of linear DAEs can be obtained. We get this result by observing that the first block row of (2.95) is a normal state-space description and showing that the solution of the second block row is a sum of the input and some of its derivatives.

Theorem 2.3
Consider a system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.100a}$$
$$y(t) = Lx(t). \tag{2.100b}$$

If (2.100) *is regular, its solution can be described by*

$$\dot{x}_1(t) = Ax_1(t) + Bu(t) \tag{2.101a}$$
$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{2.101b}$$
$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{2.101c}$$
$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.101d}$$

Proof: According to Lemma 2.3 we can without loss of generality assume that the system is in the form

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ D \end{pmatrix} u(t) \tag{2.102a}$$
$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{2.102b}$$
$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{2.102c}$$

where

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{2.103}$$

is partitioned according to the matrices.

Now, if $N = 0$ we have that

$$x_2(t) = -Du(t) \tag{2.104}$$

and we are done. If $N \neq 0$ we can multiply the second block row of (2.102a) with N to get

$$N^2\dot{x}_2(t) = Nx_2(t) + NDu(t). \tag{2.105}$$

We now differentiate (2.105) and insert the second block row of (2.102a). This gives

$$x_2(t) = -Du(t) - ND\dot{u}(t) + N^2\ddot{x}_2(t) \tag{2.106}$$

If $N^2 = 0$ we are done, otherwise we just continue until $N^m = 0$ (this is true for some m since N is nilpotent). We would then arrive at an expression like

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{2.107}$$

and the proof is complete. □

Note that the internal variables of the system, and therefore also the output, may depend directly on derivatives of the input. In the case of no dependence on the derivative of the input, we will have

$$ND = 0. \tag{2.108}$$

This relation will also play an important role in Chapter 8 where it is examined how noise can be added to the system without having to accept derivatives of the noise in the solution.

We conclude the section with an example which shows what the form (2.101) is for a simple electrical system.

Example 2.13: Canonical form

Consider the electrical circuit in Figure 2.4. With $I_1(t)$ as the output and $u(t)$ as the input,

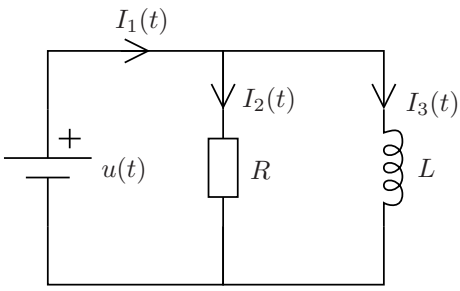


Figure 2.4: *A small electrical circuit.*

the equations describing the systems are

$$\begin{pmatrix} 0 & 0 & L \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{I}_1(t) \\ \dot{I}_2(t) \\ \dot{I}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & -R & 0 \end{pmatrix} \begin{pmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} u(t) \tag{2.109a}$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \end{pmatrix}. \tag{2.109b}$$

Transforming the system into the form (2.95) gives

$$\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}\begin{pmatrix}0&0&1\\0&1&0\\1&0&-1\end{pmatrix}\begin{pmatrix}\dot{I}_1(t)\\\dot{I}_2(t)\\\dot{I}_3(t)\end{pmatrix}=$$

$$\begin{pmatrix}0&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&1\\0&1&0\\1&0&-1\end{pmatrix}\begin{pmatrix}I_1(t)\\I_2(t)\\I_3(t)\end{pmatrix}+\begin{pmatrix}\frac{1}{L}\\-\frac{1}{R}\\-\frac{1}{R}\end{pmatrix}u(t) \tag{2.110a}$$

$$y(t)=\begin{pmatrix}1&0&0\end{pmatrix}\begin{pmatrix}I_1(t)\\I_2(t)\\I_3(t)\end{pmatrix}. \tag{2.110b}$$

Further transformation into the form (2.101) gives

$$\dot{x}_1(t)=\frac{1}{L}u(t) \tag{2.111a}$$

$$x_2(t)=-\begin{pmatrix}-\frac{1}{R}\\-\frac{1}{R}\end{pmatrix}u(t) \tag{2.111b}$$

$$\begin{pmatrix}x_1(t)\\x_2(t)\end{pmatrix}=\begin{pmatrix}0&0&1\\0&1&0\\1&0&-1\end{pmatrix}\begin{pmatrix}I_1(t)\\I_2(t)\\I_3(t)\end{pmatrix} \tag{2.111c}$$

$$y(t)=\begin{pmatrix}1&0&0\end{pmatrix}\begin{pmatrix}1&0&1\\0&1&0\\1&0&0\end{pmatrix}\begin{pmatrix}x_1(t)\\x_2(t)\end{pmatrix}. \tag{2.111d}$$

We can here see how the state-space part has been singled out by the transformation. In (2.111c) we can see that the state-space variable $x_1(t)$ is equal to $I_3(t)$. This is natural, since the only dynamic element in the circuit is the inductor. The two variables in $x_2(t)$ are $I_2(t)$ and $I_1(t) - I_3(t)$. These variables depend directly on the input.

2.3.4 Alternative Canonical Forms

The transformations presented above are the ones that will be used in this thesis, mainly because they clearly show the structure of the system and because they can be computed with numerical software as will be discussed in Chapter 11. Several other transformations have been suggested in the literature, so we will review some alternative transformations here. All methods discussed assume that the linear DAE system is regular.

Shuffle Algorithm

The *shuffle algorithm*, which was suggested by Luenberger (1978), was as the name suggests presented as an algorithm to reach a certain canonical form. The *non-reduced form*

of the shuffle algorithm applied to the DAE system (2.76) gives the canonical form

$$\dot{x}(t) = \bar{E}^{-1}\left(\bar{J}x(t) + \sum_{i=0}^{m} \bar{K}_i u^{(i)}(t)\right). \tag{2.112}$$

We show below how to calculate the matrices $\bar{E}$, $\bar{J}$, and $\bar{K}_i$. The shuffle algorithm has the advantage that no coordinate transformation is necessary. However, in (2.112) it looks as if the initial condition $x(0)$ can be chosen arbitrarily, which normally is not the case. It is instead partly determined by $u(0)$ and its derivatives. There is also a *reduced form* of the shuffle algorithm which explicitly shows how the initial conditions can be chosen.

The form (2.112) is computed by first transforming the matrix

$$\begin{pmatrix} E & J & K \end{pmatrix} \tag{2.113}$$

by row operations (for example Gauss elimination) into the form

$$\begin{pmatrix} E_1 & J_1 & K_1 \\ 0 & J_2 & K_2 \end{pmatrix} \tag{2.114}$$

where E_1 has full row rank. We now have the system

$$\begin{pmatrix} E_1 \\ 0 \end{pmatrix} \dot{x}(t) = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} x(t) + \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} u(t). \tag{2.115}$$

By differentiating the second row (this is the "shuffle" step) we get

$$\underbrace{\begin{pmatrix} E_1 \\ -J_2 \end{pmatrix}}_{\bar{E}} \dot{x}(t) = \underbrace{\begin{pmatrix} J_1 \\ 0 \end{pmatrix}}_{\bar{J}} x(t) + \underbrace{\begin{pmatrix} K_1 \\ 0 \end{pmatrix}}_{\bar{K}_0} u(t) + \underbrace{\begin{pmatrix} 0 \\ K_2 \end{pmatrix}}_{\bar{K}_1} \dot{u}(t). \tag{2.116}$$

Note that we through this differentiation loose information about the connection between the initial conditions $x(0)$ and $u(0)$. If $\bar{E}$ is non-singular, we just multiply by $\bar{E}^{-1}$ from the left to get (2.112). If it is singular, the process is continued until we get a non-singular $\bar{E}$.

SVD Coordinate System

The *SVD coordinate system* of the DAE system (2.76) is calculated by taking the singular value decomposition (SVD) of E,

$$UEV^T = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \tag{2.117}$$

where Σ contains the non-zero singular values of E and U and V are orthogonal matrices. The transformation

$$UEV^TV^{-T}\dot{x}(t) = UJV^TV^{-T}x(t) + UKu(t) \tag{2.118}$$

then gives the system

$$\begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} V^{-T}\dot{x}(t) = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} V^{-T}x(t) + \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} u(t). \tag{2.119}$$

Here, V^{-T} is the inverse of V^T. Note that $V^{-T} = V$ since V is an orthogonal matrix.

It can be noted that the block rows here do not need to have the same size as the block rows in the canonical form (2.95). The SVD coordinate system was discussed by Bender and Laub (1987) who use it to examine general system properties and to derive a linear-quadratic regulator for linear DAE systems. This transformation cannot immediately be used to get a state-space-like description, but it is used as a first step in other transformations (e.g., Kunkel and Mehrmann, 1994).

Triangular Form

We get the *triangular form* if we stay with the description in Lemma 2.1. The transformed system is then

$$\begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} Q_1^{-1}\dot{x}(t) = \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix} Q_1^{-1}x(t) + \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} u(t) \tag{2.120}$$

where E_1 is non-singular, E_3 is upper triangular with all diagonal elements zero and J_3 is non-singular and upper triangular. Using this form we could derive an expression similar to (2.101). A drawback is that here both $x_1(t)$ and $x_2(t)$ would depend on derivatives of $u(t)$, which can be verified by making calculations similar to those in the proof of Theorem 2.3. A good thing about this form is that the matrices L and R of Lemma 2.2 do not have to be computed.

2.3.5 State-Space Form

Within the control community, the theory for state-space systems is much more developed than the theory for DAE systems. For state-space systems there are many methods available for control design, state estimation and system identification, see e.g., Glad and Ljung (2000), Kailath et al. (2000), and Ljung (1999). For linear state-space systems it is also well established how the systems can be sampled, that is how an exact discrete-time counterpart of the systems can be calculated under certain assumptions on the input (e.g., Åström and Wittenmark, 1984). To be able to use these results for linear DAE systems, we in this section examine how a linear DAE system can be transformed into a linear state-space system. We will see that a linear DAE system always can be transformed to a state-space system if we are allowed to redefine the input as one of its derivatives.

What we will do is to transform a linear DAE system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.121a}$$

$$y(t) = Lx(t) \tag{2.121b}$$

into state-space form,

$$\dot{z}(t) = Az(t) + B\tilde{u}(t) \tag{2.122a}$$

$$y(t) = Cz(t) + \tilde{u}(t). \tag{2.122b}$$

Here we have written $\tilde{u}(t)$ in the state-space form to point out the fact that the input might have to be redefined as one of its derivatives. We will assume that the DAE system is regular. This implies, according to Theorem 2.3, that the system can be transformed into the form

$$\dot{x}_1(t) = Ax_1(t) + Bu(t) \tag{2.123a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{2.123b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{2.123c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.123d}$$

If $m = 1$ no derivatives of $u(t)$ occur in the description and we directly get that (2.123) is equivalent to the state-space description

$$\dot{x}_1(t) = \underbrace{A}_{\tilde{A}} x_1(t) + \underbrace{B}_{\tilde{B}} u(t) \tag{2.124a}$$

$$y(t) = \underbrace{LQ \begin{pmatrix} I \\ 0 \end{pmatrix}}_{\tilde{C}} x_1(t) + \underbrace{LQ \begin{pmatrix} 0 \\ -D \end{pmatrix}}_{\tilde{D}} u(t). \tag{2.124b}$$

If $m > 1$, the idea is to redefine the input as its $m-1$:th derivative, so the original input and some of its derivatives need to be included as state variables in the new description. We therefore define a vector with the input and some of its derivatives,

$$x_3(t) = \begin{pmatrix} u(t) \\ \dot{u}(t) \\ \vdots \\ u^{(m-2)}(t) \end{pmatrix}. \tag{2.125}$$

This vector will be part of the state vector in the transformed system. To be able to include $x_3(t)$ in the state vector, we need to calculate its derivative with respect to time:

$$\dot{x}_3(t) = \begin{pmatrix} \dot{u}(t) \\ \ddot{u}(t) \\ \vdots \\ u^{(m-1)}(t) \end{pmatrix} = \begin{pmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} x_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u^{(m-1)}(t) \tag{2.126}$$

We can now rewrite (2.123) to depend on $x_3(t)$ instead of depending directly on the

different derivatives of $u(t)$. The new description will be

$$\dot{x}_1(t) = Ax_1(t) + \begin{pmatrix} B & 0 & \dots & 0 \end{pmatrix} x_3(t) \tag{2.127a}$$

$$x_2(t) = -\begin{pmatrix} D & ND & \dots & N^{m-2}D \end{pmatrix} x_3(t) - N^{m-1}Du^{(m-1)}(t) \tag{2.127b}$$

$$\dot{x}_3(t) = \begin{pmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} x_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u^{(m-1)}(t) \tag{2.127c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{2.127d}$$

The final step to obtain a state-space description is to eliminate $x_2(t)$ from these equations. The elimination is performed by inserting (2.127b) into (2.127d):

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_3(t) \end{pmatrix} = \underbrace{\begin{pmatrix} A & B & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{pmatrix}}_{\tilde{B}} u^{(m-1)}(t) \tag{2.128a}$$

$$\begin{aligned} y(t) = {} & \underbrace{LQ \begin{pmatrix} I & 0 & 0 & \dots & 0 \\ 0 & -D & -ND & \dots & -N^{m-2}D \end{pmatrix}}_{\tilde{C}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} \\ & + \underbrace{LQ \begin{pmatrix} 0 \\ -N^{m-1}D \end{pmatrix}}_{\tilde{D}} u^{(m-1)}(t) \end{aligned} \tag{2.128b}$$

If we let

$$z(t) = \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} \tag{2.129}$$

this can be written in the compact form

$$\dot{z}(t) = \tilde{A}z(t) + \tilde{B}u^{(m-1)}(t) \tag{2.130a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}u^{(m-1)}(t). \tag{2.130b}$$

The main purpose of this thesis is to examine how unknown parameters and internal variables in DAE systems can be estimated, and this is what the state-space system will be used for in the following. However, as pointed out in the beginning, it may be useful to do the conversion in other cases as well, for example when designing controllers. The controller would then generate the control signal $u^{(m-1)}(t)$. In order to obtain the actual control signal $u(t)$ we have to integrate $u^{(m-1)}(t)$. For a further discussion on this, see e.g., the paper by Müller (2000).

We conclude the section by continuing Example 2.13 and writing the system in state-space form.

Example 2.14: State-space form

In Example 2.13 we saw that the equations for the electrical circuit could be written as

$$\dot{x}_1(t) = \frac{1}{L}u(t) \tag{2.131a}$$

$$x_2(t) = -\begin{pmatrix} -\frac{1}{R} \\ -\frac{1}{R} \end{pmatrix} u(t) \tag{2.131b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \end{pmatrix} \tag{2.131c}$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.131d}$$

Since $m = 1$ (no derivatives of $u(t)$ occur in the description), $x_3(t)$ is not necessary and (2.124) can be used. This gives us the state-space description

$$\dot{z}(t) = \frac{1}{L}u(t) \tag{2.132a}$$

$$y(t) = z(t) + \frac{1}{R}u(t). \tag{2.132b}$$

For this simple case, the state-space description could have been derived manually from the original equations, but the procedure in the example shows how we can compute the state-space description automatically. For larger systems it may be more difficult to derive the state-space description manually.

2.3.6 Sampling

As discussed earlier, the theory for state-space systems is much more developed than the theory for DAE systems. In the previous section, we showed how a linear DAE system can be transformed into a continuous-time state-space system, which gives us the possibility to use theory for continuous-time state-space systems. However, in many cases measured data from a system is available as sampled data. This could be the case both for control, for estimation, and for system identification. To handle such cases for continuous-time state-space systems, one common approach is to *sample* the state-space system, that is to calculate a discrete-time counterpart of the state-space system. In this section we examine how a linear DAE system can be sampled.

The basic result for sampling of state-space systems with piecewise constant input is given in Lemma 2.4 below. The main result of this section is the extension of this lemma to linear DAE systems.

Lemma 2.4

Consider the state-space system

$$\dot{z}(t) = Az(t) + Bu(t) \tag{2.133a}$$

$$y(t) = Cz(t) + Du(t). \tag{2.133b}$$

If $u(t)$ is constant for $T_s k \leq t < T_s k + T_s$ for constant T_s and $k = 0, 1, 2, ...$, then $z(T_s k)$ and $y(T_s k)$ are exactly described by the discrete-time state-space system

$$z(T_s k + T_s) = \Phi z(T_s k) + \Gamma u(T_s k) \tag{2.134a}$$

$$y(T_s k) = C z(T_s k) + D u(T_s k), \tag{2.134b}$$

where

$$\Phi = e^{AT_s} \tag{2.135}$$

$$\Gamma = \int_0^{T_s} e^{A\tau} d\tau B. \tag{2.136}$$

Proof: See, for example, the book by Åström and Wittenmark (1984). □

Now, if we assume that $u^{(m-1)}(t)$ is piecewise constant, Lemma 2.4 can be applied to (2.124) or (2.130) to give an exact discrete-time description of the original linear DAE system. We have thus arrived at the following theorem:

Theorem 2.4
Consider the regular linear DAE system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{2.137a}$$

$$y(t) = Lx(t) \tag{2.137b}$$

with the canonical form

$$\dot{x}_1(t) = Ax_1(t) + Bu(t) \tag{2.138a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{2.138b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1} x(t) \tag{2.138c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.138d}$$

If $u^{(m-1)}(t)$ is constant for $T_s k \leq t < T_s k + T_s$ for constant T_s and $k = 0, 1, 2, ...$, then $y(T_s k)$ is exactly described by the discrete-time state-space system

$$z(T_s k + T_s) = \Phi z(T_s k) + \Gamma u^{(m-1)}(T_s k) \tag{2.139a}$$

$$y(T_s k) = \tilde{C} z(T_s k) + \tilde{D} u^{(m-1)}(T_s k). \tag{2.139b}$$

where

$$\Phi = e^{\tilde{A}T_s} \tag{2.140}$$

$$\Gamma = \int_0^{T_s} e^{\tilde{A}\tau} d\tau \tilde{B} \tag{2.141}$$

and $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ are defined in (2.124) *or* (2.128).

Note that there are other assumptions on the behavior of $u^{(m-1)}(t)$ between the sample points which also will allow us to calculate an exact discrete-time description. One such assumption is that it is piecewise linear.

If the internal variables do not depend on derivatives of the input, we will have $ND = 0$ in the equations above. The derivations in this section are of course valid also for this case, although many of the formulas can be written in a simpler form. For example, we will have $m = 1$, so we do not need to redefine the input. However, note that the matrix E in the linear DAE system may very well be singular even if there is no dependence on derivatives of the input, so it is still advantageous to use the formulas above to write the system in state-space form and sample it.

For state-space systems, it is typically assumed that the input (and not one of its derivatives) is piecewise constant when sampling a system. For DAEs where the internal variables depend on derivatives of the input, this is not a realistic assumption since the internal variables would be derivatives of a step function.

2.4 Linear Time-Varying DAE Models

A more general form of the linear DAE is the linear time-varying DAE,

$$E(t)\dot{x}(t) = A(t)x(t) + f(t) \tag{2.142a}$$
$$y(t) = C(t)x(t). \tag{2.142b}$$

$E(t)$ and $A(t)$ are square time-varying matrices and $C(t)$ is a rectangular time-varying matrix. The external function $f(t)$ typically represents an input signal which for example can enter the equations as $f(t) = B(t)u(t)$ where $u(t)$ is an input signal. We will stick with the notation $f(t)$ in this section to be consistent with the notation by Kunkel and Mehrmann (1994) from which we will present some results. (Time-varying linear DAE models are also discussed in the book by Kunkel and Mehrmann, 2006). The results from Kunkel and Mehrmann (1994) which we will review here treat canonical forms for linear time-varying DAE systems. The canonical forms of are, not surprisingly, similar to the time-invariant case. The main difference is that time-varying transformations are used, that is (2.142a) is multiplied from the left with a matrix $P(t)$ and a variable transformation $x(t) = Q(t)\tilde{x}(t)$ is made. Since

$$\dot{x}(t) = \dot{Q}(t)\tilde{x}(t) + Q(t)\dot{\tilde{x}}(t) \tag{2.143}$$

the transformed system is

$$P(t)E(t)Q(t)\dot{\tilde{x}}(t) = \big(P(t)A(t)Q(t) - P(t)E(t)\dot{Q}(t)\big)\tilde{x}(t) + P(t)f(t) \tag{2.144a}$$
$$y(t) = C(t)Q(t)x(t). \tag{2.144b}$$

We see that there is an extra term $P(t)E(t)\dot{Q}(t)$ in (2.144a) compared to the time-invariant case. This makes the transformations somewhat more involved.

First we will need to define a few quantities related to the $n \times n$ matrices $E(t)$

and $A(t)$. Let

$$T(t) \text{ basis of } \operatorname{kernel} E(t) \tag{2.145a}$$
$$Z(t) \text{ basis of } \operatorname{corange} E(t) \tag{2.145b}$$
$$T'(t) \text{ basis of } \operatorname{cokernel} E(t) \tag{2.145c}$$
$$V(t) \text{ basis of } \operatorname{corange}\big(Z^*(t)A(t)T(t)\big). \tag{2.145d}$$

A^* is the conjugate transpose of the matrix A. The kernel (or null space), range, corange, and cokernel of an $n \times n$ matrix A are defined as

$$\begin{aligned}
\operatorname{kernel} A &= \{y \in \mathbb{R}^n | Ay = 0\} \\
\operatorname{range} A &= \{y \in \mathbb{R}^n | y = Ax, x \in \mathbb{R}^n\} \\
\operatorname{corange} A &= \operatorname{kernel} A^* \\
\operatorname{cokernel} A &= \operatorname{range} A^*.
\end{aligned}$$

Now, let

$$r(t) = \operatorname{rank} E(t) \tag{2.146a}$$
$$a(t) = \operatorname{rank}\big(Z^*(t)A(t)T(t)\big) \tag{2.146b}$$
$$s(t) = \operatorname{rank}\big(V^*(t)Z^*(t)A(t)T'(t)\big) \tag{2.146c}$$
$$d(t) = r(t) - s(t) \tag{2.146d}$$
$$b(t) = n - r(t) - a(t) - s(t). \tag{2.146e}$$

The quantities $r(t)$, $a(t)$, $s(t)$, $d(t)$, and $b(t)$ are called rank, algebraic part, strangeness, differential part, and undetermined part respectively. We can now state the main transformation theorem from Kunkel and Mehrmann (1994).

Theorem 2.5
Let the matrices $E(t)$ and $A(t)$ be sufficiently smooth and let

$$r(t) \equiv r \tag{2.147a}$$
$$a(t) \equiv a \tag{2.147b}$$
$$s(t) \equiv s. \tag{2.147c}$$

Then there exist non-singular transformation matrices $P(t)$ and $Q(t)$ such that

$$P(t)E(t)Q(t) = \begin{pmatrix} I_s & 0 & 0 & 0 & 0 \\ 0 & I_d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.148}$$

and

$$P(t)A(t)Q(t) - P(t)E(t)\dot{Q}(t) = \begin{pmatrix} 0 & A_{12}(t) & 0 & A_{14}(t) & A_{15}(t) \\ 0 & 0 & 0 & A_{24}(t) & A_{25}(t) \\ 0 & 0 & I_a & 0 & 0 \\ I_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.149}$$

I_m is an identity matrix of size $m \times m$. The sizes of the block rows are s, d, a, s, and b respectively.

Proof: See Kunkel and Mehrmann (1994). □

Note that this transformation means that the system in the transformed variables

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{pmatrix} = Q(t)x(t) \tag{2.150}$$

can be written as

$$\dot{x}_1(t) = A_{12}(t)x_2(t) + A_{14}(t)x_4(t) + A_{15}(t)x_5(t) + f_1(t) \tag{2.151a}$$
$$\dot{x}_2(t) = A_{24}(t)x_4(t) + A_{25}(t)x_5(t) + f_2(t) \tag{2.151b}$$
$$0 = x_3(t) + f_3(t) \tag{2.151c}$$
$$0 = x_1(t) + f_4(t) \tag{2.151d}$$
$$0 = f_5(t) \tag{2.151e}$$

where

$$\begin{pmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \\ f_4(t) \\ f_5(t) \end{pmatrix} = P(t)f(t). \tag{2.152}$$

The form (2.151) can be further transformed by differentiating (2.151d) and inserting into (2.151a). We then get

$$0 = A_{12}(t)x_2(t) + A_{14}(t)x_4(t) + A_{15}(t)x_5(t) + f_1(t) + \dot{f}_4(t) \tag{2.153a}$$
$$\dot{x}_2(t) = A_{24}(t)x_4(t) + A_{25}(t)x_5(t) + f_2(t) \tag{2.153b}$$
$$0 = x_3(t) + f_3(t) \tag{2.153c}$$
$$0 = x_1(t) + f_4(t) \tag{2.153d}$$
$$0 = f_5(t). \tag{2.153e}$$

This form can the be further transformed by applying Theorem 2.5 again. Repeating this process until $s = 0$ for the transformed system (or equivalently, the size of $x_1(t)$ is equal to zero) leads to the following theorem (Kunkel and Mehrmann, 1994).

Theorem 2.6
Let $E_i(t)$, $A_i(t)$ be the sequence of matrices that is obtained by repeatedly applying Theorem 2.5 and differentiating to obtain the form (2.153) *for a linear time-varying DAE* (2.142a). *Let $s_i(t)$ be the strangeness for each pair of matrices. Let the strangeness index defined by*

$$m = \min\{i | s_i = 0\} \tag{2.154}$$

be well determined. Let the function $f(t)$ be sufficiently differentiable. Then (2.142a) *is equivalent to a differential-algebraic equation in the form*

$$\dot{x}_1(t) = A_{13}(t)x_3(t) + f_1(t) \tag{2.155a}$$

$$0 = x_2(t) + f_2(t) \tag{2.155b}$$

$$0 = f_3(t). \tag{2.155c}$$

Here, $f_1(t)$ is determined by the function $f(t)$, and $f_2(t)$ and $f_3(t)$ are determined by $f(t)$, $\dot{f}(t)$, ..., $f^{(m)}(t)$.

Proof: See Kunkel and Mehrmann (1994). □

Equivalence here means that there is a one-to-one relationship between the solutions. The fact that $f_1(t)$ does not depend on derivatives of $f(t)$ is not directly stated by Kunkel and Mehrmann (1994), but it is given by the transformations involved. Each transformation has matrices $Q_i(t)$ and $P_i(t)$ such that

$$x(t) = Q_1(t)Q_2(t)\cdots Q_m(t)\begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} \tag{2.156}$$

and

$$\begin{pmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \end{pmatrix} = P_{m+1}(t)\tilde{P}_m(t, \tfrac{d}{dt})\cdots\tilde{P}_1(t, \tfrac{d}{dt})f(t) \tag{2.157}$$

where

$$\tilde{P}_i = \begin{pmatrix} I & 0 & 0 & \frac{d}{dt}I & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} P_i(t). \tag{2.158}$$

The matrix containing $\frac{d}{dt}$ represents the differentiation that took us to the form (2.153). Note that no differentiation is performed in the final transformation $P_{m+1}(t)$ since $s_i = 0$. It is not apparent from this notation that $f_1(t)$ does not depend on derivatives of f, but this is given by the proofs in Kunkel and Mehrmann (1994) where block rows containing differentiated variables $\dot{x}(t)$ are not mixed with rows not containing differentiated variables.

Theorem 2.6 makes it possible to define regularity for linear time-varying DAEs as the absence of undetermined variables $x_3(t)$.

Definition 2.8 (Regularity). The linear time-varying DAE is said to be *regular* if there are no undetermined variables, or equivalently, $x_3(t)$ in (2.155) is of size zero.

2.5 DAE Solvers

This section introduces the basic functionality of DAE solvers, and assumptions about the solver that will be needed to derive some of the results in the thesis. The basic functionality that is assumed is that given a nonlinear DAE, $F(\dot{x}(t), x(t), t) = 0$, the solvers produce $x(t)$ for a desired time interval.

Purely numeric solvers for DAEs only handle limited classes of DAEs, usually systems with differential index 1, or limited classes of higher index systems. One common numerical solver is DASSL (Brenan et al., 1996). For component-based models, it is not sufficient to treat lower index problems, so instead the kind of solvers that are used to simulate component-based models, such as Modelica models, are used. Such solvers are included in, e.g., Dymola and OpenModelica. These solvers typically reduce the index to 1 by differentiating equations that are chosen using Pantelides's algorithm (Pantelides, 1988) and structure the equations so that large DAE systems can be simulated efficiently. Then a numerical solver is used. The number of times that

Pantelides's algorithm (Pantelides, 1988) is an important tool for finding which equations to differentiate when reducing the index of large-scale higher index DAE systems. This is a graph-theoretical algorithm that originally was developed to find conditions that consistent initial values must satisfy. It has later been used by others to find differentiations to reduce the index of DAE systems to 1 or 0 in DAE solvers. The algorithm only uses structural information about which variables that are included in which equations. While the algorithm works well for index reduction in most cases, it can sometimes give incorrect results (Reißig et al., 2000).

Structuring of the equations to achieve efficient simulation can be performed by transforming the equations into block lower triangular (BLT) form. This means that the equations are sorted so that the equations can be solved stepwise for a few variables at a time. An implementation of the BLT algorithm (not in connection with equation solving) is discussed by Duff and Reid (1978).

During the index reduction process, some of the variables $x(t)$ are selected as states. For the user, this means that initial values of these variables can be selected independently of each other. The initial values of the remaining variables are computed from the initial values of the states so that the initial value is consistent. It is possible for the user to influence the state selection process by indicating that some variables are preferred as states.

The solver typically also structures the equations as

$$\tilde{F}_1(t, x_1, x_3, \dot{x}_1) = 0 \tag{2.159a}$$

$$\hat{F}_2(t, x_1, x_3) = 0 \tag{2.159b}$$

where x_3 can be solved from (2.159b) and $\dot{x}_1$ can be solved from (2.159a). This means that an approximation of the transformations discussed in Section 2.2 is computed.

One method to combine these methods into an index reducing pre-processing algorithm is suggested by Mattsson and Söderlind (1993). We summarize this algorithm here.

1. Differentiate the equations using Pantelides's algorithm to achieve an index 1 system.

2. Permute the equations and variables into BLT form.

3. Select state variables using the *dummy derivatives method* (Mattsson and Söderlind, 1993).

To derive some of the identifiability results later in the thesis, some assumptions on the DAE solver are needed:

- If locally unique solutions to the DAE exist, one of them is given. Otherwise the user is notified, e.g., through an error message. There are different error messages for the cases when no solution exists and when existing solutions are not locally unique.

- Some of the variables that appear differentiated are selected as states by the solver. The initial values of these variables can be selected freely by the user, and the initial values of the remaining variables are computed from the initial values of the state variables.

- The number of unknowns must be the same as the number of equations. (The derivative of a variable does not count as an unknown of its own.)

These assumptions represent a kind of ideal DAE solver, but existing DAE solvers such as the one in Dymola come quite close to satisfying them.

2.6 Linear Difference-Algebraic Equations

In this section the difference-algebraic system

$$Ex(t+1) = Jx(t) + Ku(t) \tag{2.160a}$$
$$y(t) = Lx(t) \tag{2.160b}$$

will be treated. Difference-algebraic equations are also known as discrete-time descriptor systems. Since the sampled version of a linear DAE system can be written as a discrete-time state-space system (see Section 2.3.6), there are probably fewer applications for discrete-time descriptor systems than for discrete-time state-space systems. However, applications could be found among truly discrete-time systems such as some economical systems. Discrete-time and continuous-time descriptor systems can be treated in a similar fashion, so the discussion here will be rather brief.

We will show how (2.160) can be written in different canonical forms and then transformed into state-space form, but we can directly note that (2.160) is a discrete-time linear system with the transfer function

$$G(z) = L(zE - J)^{-1}K. \tag{2.161}$$

A difference between $G(z)$ and the transfer function of a discrete-time state-space system is that $G(z)$ here may be non-proper, that is have higher degree in the numerator than in the denominator. This corresponds to a non-causal system. For an example of matrices E and J that give a non-proper system, see (2.79).

Similarly to the continuous-time case, the transfer function is only well-defined if $(zE - J)$ is non-singular. In the next section we will define non-singularity of this matrix as regularity for the corresponding system and show that the system is solvable if the system is regular.

2.6.1 Regularity

A basic assumption that will be made about the discrete-time descriptor systems is that the inverse in (2.161) is well-defined, and below this is formalized with a definition.

Definition 2.9 (Regularity). The discrete-time descriptor system

$$Ex(t+1) = Jx(t) + Ku(t) \tag{2.162a}$$
$$y(t) = Lx(t) \tag{2.162b}$$

is called *regular* if

$$\det(zE - J) \not\equiv 0, \tag{2.163}$$

that is the determinant is not zero for all z.

This definition is the same as the one used by Dai (1989b). As in the continuous-time case, regularity is equivalent to the existence of unique solution. This is discussed by for example Luenberger (1978) and Dai (1989b). To illustrate this we examine the z transform of equation (2.162a):

$$(zE - J)\,\mathcal{Z}\left[x(t)\right] = K\mathcal{Z}\left[u(t)\right] + zEx(0) \tag{2.164}$$

$\mathcal{Z}[\cdot]$ represents the z transform of the argument. From this equation we can draw the conclusion that there exists a unique solution $\mathcal{Z}[x(t)]$ almost everywhere if and only if the system is regular.

2.6.2 A Canonical Form

In this section we present a transformation for discrete-time descriptor systems, which gives a canonical form similar to the one for the continuous-time case presented in Section 2.3.3. The only difference between the two forms is actually that the derivatives in the continuous-time case are replaced by time shifts in the discrete-time case.

Theorem 2.7
Consider a system

$$Ex(t+1) = Jx(t) + Ku(t) \tag{2.165a}$$
$$y(t) = Lx(t). \tag{2.165b}$$

If (2.165) *is regular, its solution can be described by*

$$x_1(t+1) = Ax_1(t) + Bu(t) \tag{2.166a}$$
$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i Du(t+i) \tag{2.166b}$$
$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{2.166c}$$
$$y(t) = LQ\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.166d}$$

The proof is the same as the one for Theorem 2.3 with all derivatives replace with time shifts (also in the required lemmas), so it is omitted.

Note that the system is non-causal in the general case as the output can depend on future values of the input. However, if

$$ND = 0 \tag{2.167}$$

the system is causal.

2.6.3 State-Space Form

As mentioned earlier, state-space systems are much more thoroughly treated in the literature than descriptor systems are. This is also true for the discrete-time case, so in this section we examine how a discrete-time descriptor system can be transformed to a discrete-time state-space system.

We assume that the system has been converted into the form

$$x_1(t+1) = Ax_1(t) + Bu(t) \tag{2.168a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i Du(t+i) \tag{2.168b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{2.168c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \tag{2.168d}$$

which according to Theorem 2.7 is possible if the system is regular. If $m = 1$, we directly get the state-space description

$$x_1(t+1) = \underbrace{A}_{\tilde{A}} x_1(t) + \underbrace{B}_{\tilde{B}} u(t) \tag{2.169a}$$

$$y(t) = \underbrace{LQ \begin{pmatrix} I \\ 0 \end{pmatrix}}_{\tilde{C}} x_1(t) + \underbrace{LQ \begin{pmatrix} 0 \\ -D \end{pmatrix}}_{\tilde{D}} u(t). \tag{2.169b}$$

If $m > 1$ we begin by defining a vector with time shifted inputs, corresponding to Equation (2.125):

$$x_3(t) = \begin{pmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+m-2) \end{pmatrix} \tag{2.170}$$

To include $x_3(t)$ in the state vector, the time shifted version of it must be calculated:

$$x_3(t+1) = \begin{pmatrix} u(t+1) \\ u(t+2) \\ \vdots \\ u(t+m-1) \end{pmatrix} = \begin{pmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} x_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u(t+m-1) \tag{2.171}$$

Now (2.168) can be rewritten to depend on $x_3(t)$ instead of depending directly on the time shifted versions of $u(t)$. The new description of the solutions will be

$$x_1(t+1) = Ax_1(t) + \begin{pmatrix} B & 0 & \dots & 0 \end{pmatrix} x_3(t) \tag{2.172a}$$

$$\begin{aligned} x_2(t) = &- \begin{pmatrix} D & ND & \dots & N^{m-2}D \end{pmatrix} x_3(t) \\ &- N^{m-1}Du(t+m-1) \end{aligned} \tag{2.172b}$$

$$x_3(t+1) = \begin{pmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} x_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u(t+m-1) \tag{2.172c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{2.172d}$$

The final step to get a state-space description is to eliminate $x_2(t)$ from these equations. The elimination is performed by inserting (2.172b) into (2.172d):

$$\begin{pmatrix} x_1(t+1) \\ x_3(t+1) \end{pmatrix} = \underbrace{\begin{pmatrix} A & B & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{pmatrix}}_{\tilde{B}} u(t+m-1) \tag{2.173a}$$

$$\begin{aligned} y(t) = &\underbrace{LQ \begin{pmatrix} I & 0 & 0 & \dots & 0 \\ 0 & -D & -ND & \dots & -N^{m-2}D \end{pmatrix}}_{\tilde{C}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} + \\ &\underbrace{LQ \begin{pmatrix} 0 \\ -N^{m-1}D \end{pmatrix}}_{\tilde{D}} u(t+m-1) \end{aligned} \tag{2.173b}$$

If we let

$$z(t) = \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} \tag{2.174}$$

this can be written in the compact form

$$z(t+1) = \tilde{A}z(t) + \tilde{B}u(t+m-1) \tag{2.175a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}u(t+m-1). \tag{2.175b}$$

The state-space description will in this thesis be used for estimation. However, it could also have other applications, such as control design.

2.7 Stochastic Models

As discussed in the introduction of the chapter, it is often appropriate to model disturbances using stochastic processes. This section summarizes some results on the definition

and properties of stochastic processes that will be needed for the discussions later in the thesis. Most of the results summarized here can be found in, e.g., the books by Åström (1970) and Papoulis (1977).

2.7.1 Stochastic Processes

A stochastic process can be defined as a family of stochastic variables indexed by a set T, $\{x(t), t \in T\}$. The set T will be interpreted as time in this thesis. When it takes discrete values, $T = \{\dots, -1, 0, 1, \dots\}$ or $T = \{0, 1, \dots\}$, the process x is called a *discrete-time process*. When T takes continuous values, $T = \{t; -\infty < t < \infty\}$ or $T = \{t; 0 \leq t < \infty\}$, the process is called a *continuous-time process*.

We need to define a number of properties for a stochastic process. The *mean value* is defined as

$$m(t) = \mathrm{E}\, x(t). \tag{2.176}$$

Furthermore, the *covariance function* for the processes $\{x(t), t \in T\}$ and $\{y(t), t \in T\}$ is defined as

$$r_{xy}(s,t) = \operatorname{cov}\big(x(s), y(t)\big) = \mathrm{E}\big(x(s) - \mathrm{E}\, x(s)\big)\big(y(t) - \mathrm{E}\, y(t)\big)^T. \tag{2.177}$$

If $\{x(t), t \in T\}$ and $\{y(t), t \in T\}$ are the same, the function $r_{xx}(s,t)$ is called the *autocovariance function*. It is then also denoted $r_x(s,t)$, or simply $r(s,t)$ when it is clear to which process it belongs. The variance of a stochastic process is

$$\operatorname{var}\big(x(t)\big) = r_x(t,t). \tag{2.178}$$

A process $\{x(t), t \in T\}$ is said to be of *second order* if $\mathrm{E}\, x^2(t) < \infty$ for all $t \in T$.

A stochastic process is said to be *Gaussian* (or *normal*) if the joint distribution of $\big(x(t_1), x(t_2), \dots, x(t_k)\big)$ is Gaussian for every k and all $t_i \in T$, $i = 1, 2, \dots, k$. A Gaussian process is completely characterized by its mean value and covariances.

A process is said to be *stationary* if the distribution of $\big(x(t_1), x(t_2), \dots, x(t_k)\big)$ is the same as the distribution of $\big(x(t_1 + \tau), x(t_2 + \tau), \dots, x(t_k + \tau)\big)$ for all $t_i \in T$ and all τ such that $t_i + \tau \in T$. A process is said to be *weakly stationary* if the mean values and covariances but not necessarily the distributions are the same. Note especially that the covariance function can be written $r(s,t) = r(s-t)$ for weakly stationary processes.

A process $\{x(t), t \in T\}$ where $x(t_k) - x(t_{k-1})$, $x(t_{k-1}) - x(t_{k-2})$, $\dots$, $x(t_2) - x(t_1)$, $x(t_1)$ for $t_1 < t_2 < \dots < t_k$ are mutually independent is called *a process with independent increments*. Processes with independent increments can be used to define a *Wiener process* or a *Brownian motion process* which is a process which satisfies the following conditions:

1. $x(0) = 0$
2. $x(t)$ is Gaussian
3. $\mathrm{E}\, x(t) = 0$ for all $t > 0$
4. The process has independent stationary increments

The *spectral density function* or *spectral density* $\phi(\omega)$ of a weakly stationary process describes its frequency content. In this thesis it is also called *spectrum*. For a continuous-time process with autocovariance function $r(t)$ it is defined as

$$\phi(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega t} r(t)\, dt \tag{2.179a}$$

$$r(t) = \int_{-\infty}^{\infty} e^{i\omega t} \phi(\omega)\, d\omega \tag{2.179b}$$

and for a discrete-time process with autocovariance function $r(t)$ it is defined as

$$\phi(\omega) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} e^{-i\omega n} r(n) \tag{2.180a}$$

$$r(n) = \int_{-\pi}^{\pi} e^{i\omega n} \phi(\omega)\, d\omega. \tag{2.180b}$$

When r is a covariance function for two processes x and y, r_{xy}, ϕ_{xy} is called the *cross spectral density*.

A weakly stationary process with constant spectral density $\phi(\omega) \equiv \phi$ is called *white noise*. This definition applies both for discrete-time and continuous-time processes. White noise for continuous-time processes requires a more involved analysis than the discrete-time case, but we will not go into the details here. That there are special problems with continuous-time white noise can for example be realized from (2.179b) which gives that the variance $r(t)$ is infinite if $\phi(\omega)$ is constant. The reader is referred to, e.g., the book by Åström (1970) for further discussions on this.

2.7.2 Continuous-Time Linear Stochastic Models

As discussed in Section 2.1.2, we would like to define a stochastic differential equation, SDE, according to

$$\dot{x}(t) = Ax(t) + Kv(t) \tag{2.181}$$

where $v(t)$ is a stochastic process. We here omit the deterministic input $u(t)$ in the notation since it does not affect the results discussed here. When $\{v(t), t \in T\}$ is continuous-time white noise with spectrum R_1, which also can be seen as covariance $2\pi R_1 \delta(t)$, $v(t)$ has infinite variance. This means that $\dot{x}(t)$ would not be well-defined. We instead have to interpret the expression (2.181) with $v(t)$ white noise as a stochastic integral (Åström, 1970). To point this out, the notation

$$dx = Axdt + Kdv \tag{2.182}$$

where $\{v(t), t \in T\}$ is a Wiener process with incremental covariance $R_1 dt$ can be used.

The solution of the stochastic integral can be interpreted booth as an *Itô integral* and as a *Stratonovich integral* (Åström, 1970). Irrespective of which integral concept that is used, the solution is characterized by the following theorem.

Theorem 2.8
Assume that the initial value $x(t_0)$ of the stochastic differential equation (2.182) *is a Gaussian stochastic variable with mean m_0 and covariance matrix R_0 and that $v(t)$ is a Wiener process with incremental covariance $R_1 dt$. The solution of the SDE is then a normal process with mean value $m_x(t)$ and covariance $R(s,t)$ where*

$$\frac{dm_x}{dt} = Am_x \tag{2.183a}$$

$$m_x(t_0) = m_0 \tag{2.183b}$$

$$R(s,t) = \begin{cases} \Phi(s;t)P(t) & s \geq t \\ P(s)\Phi^T(t;s) & s \leq t \end{cases} \tag{2.183c}$$

$$\frac{dP}{dt} = AP + PA^T + KR_1K^T \tag{2.183d}$$

$$P(t_0) = R_0 \tag{2.183e}$$

$$\frac{d\Phi(t;t_0)}{dt} = A\Phi(t;t_0) \tag{2.183f}$$

$$\Phi(t_0,t_0) = I. \tag{2.183g}$$

Proof: See Åström (1970). □

We would also like to use a transfer function description of a stochastic process, that is to write a stochastic process $\{y(t), t \in T\}$ as

$$y(t) = G(p)w(t) \tag{2.184}$$

where $G(p)$ is a transfer function and $\{w(t), t \in T\}$ is a stochastic process with spectrum $\phi_w(\omega)$. This can also be written as a convolution integral,

$$y(t) = \int_{-\infty}^{t} h(s-t)w(s)\,ds \tag{2.185}$$

where $h(t)$ is the impulse response corresponding to the transfer function $G(p)$. If $w(t)$ has finite variance, i.e., it is not white noise, this integral has a well-defined solution, which is given in the following theorem.

Theorem 2.9
Suppose that a time-invariant dynamical system has the scalar transfer function G. Suppose that the input signal is a weakly stationary stochastic process with mean value m_w and spectral density $\phi_w(\omega)$. If the dynamical system is asymptotically stable and if

$$r_w(0) = \int_{-\infty}^{\infty} \phi_w(\omega)\,d\omega \leq a < \infty \tag{2.186}$$

then the output signal (after transients have disappeared) is a weakly stationary process with mean value

$$m_y = G(0)m_w \tag{2.187}$$

and spectral density

$$\phi_y(\omega) = G(i\omega)G(-i\omega)\phi_w(\omega). \tag{2.188}$$

The input-output cross spectral density is

$$\phi_{wy}(\omega) = G(-i\omega)\phi_w(\omega). \tag{2.189}$$

Proof: See Åström (1970). □

If $w(t)$ is to be interpreted as white noise in (2.184), then (2.184) must be seen as shorthand notation for the stochastic integral

$$y(t) = \int_{\infty}^{t} h(t-s)\, dw(s) \tag{2.190}$$

where h is the impulse response of the transfer function G and $\{w(t), t \in T\}$ is a Wiener process with orthogonal increments. If we want to select a linear filter so that $y(t)$ has a certain spectrum, the following theorem can be used (Åström, 1970).

Theorem 2.10
Consider a rational spectral density function $\phi(\omega)$. There exists an asymptotically stable, time invariant dynamical system with the impulse response h such that the stochastic process defined by

$$y(t) = \int_{\infty}^{t} h(t-s)\, dw(s) \tag{2.191}$$

where $\{w(t), t \in T\}$ is a process with independent increments, is stationary and has the spectral density $\phi(\omega)$. Furthermore, if w has incremental covariance $2\pi dt$, the transfer function G corresponding to h can be chosen as

$$\phi(\omega) = G(i\omega)G(-i\omega) \tag{2.192}$$

with all poles in the left half plane and all zeros in the left half plane or on the imaginary axis.

Proof: See Åström (1970). □

2.7.3 Discrete-Time Linear Stochastic Models

It is easier to define a stochastic difference equation than a stochastic differential equation. A linear stochastic difference equation can be written as

$$x(t+1) = Ax(t) + Kv(t) \tag{2.193}$$

where the process $\{v(t), t \in T\}$ is discrete-time white noise. When $v(t)$ is Gaussian with mean value zero and covariance matrix R_1, the solution is characterized by the following theorem (Åström, 1970).

Theorem 2.11
The solution of the stochastic difference equation (2.193) *where the initial value is a Gaussian random variable with mean* m_0 *and covariance matrix* R_0 *and* $v(t)$ *is Gaussian with mean value zero and covariance* R_1*, is a Gaussian random process with the mean value*

$$m(t+1) = Am(t) \tag{2.194}$$

with the initial condition

$$m(t_0) = m_0 \tag{2.195}$$

and the covariance function

$$R(s,t) = A^{s-t}P(t) \quad s \geq t \tag{2.196}$$

where $P(t)$ *satisfies*

$$P(t+1) = AP(t)A^T + KR_1K^T \tag{2.197}$$

with the initial condition

$$P(t_0) = R_0. \tag{2.198}$$

Proof: See Åström (1970). □

Also the transfer function description is easier to handle for discrete-time systems than for continuous-time systems. Consider the description

$$y(t) = H(q)u(t) \tag{2.199}$$

where $\{u(t), t \in T\}$ is a stationary process. This can also be written as a convolution,

$$y(t) = \sum_{s=-\infty}^{t} h(t-s)u(s). \tag{2.200}$$

The solution is characterized by the following theorem (Åström, 1970).

Theorem 2.12
Consider a stationary discrete-time system with the transfer function $H(z)$*. Let the input signal be a stationary stochastic process with mean value* m_u *and the spectral density* $\phi_u(\omega)$*. If the system is asymptotically stable, then the output signal is a stationary stochastic process with mean value*

$$m_y = H(1)m_u \tag{2.201}$$

and spectral density

$$\phi_y(\omega) = H(e^{-i\omega})H(e^{i\omega})\phi_u(\omega) = |H(e^{i\omega})|^2\phi_u(\omega). \tag{2.202}$$

The cross spectral density between input and output is given by

$$\phi_{yu}(\omega) = H(e^{-i\omega})\phi_u(\omega) \tag{2.203}$$

Proof: See Åström (1970). □

Note that this theorem also holds for white noise inputs, that is constant $\phi_u(\omega)$.

It is possible to transform a continuous-time stochastic differential equation for which the output is measured at sampling instants into a discrete-time system. The following lemma describes how this transformation is performed (Åström, 1970).

Lemma 2.5
Consider a state-space system with noise model

$$dx = Axdt + dv \tag{2.204a}$$
$$dy = Cxdt + de \tag{2.204b}$$

where v and e are Wiener processes with incremental covariances

$$\mathrm{E}\left(dv\, dv^T\right) = R_1 dt \tag{2.205a}$$
$$\mathrm{E}\left(dv\, de^T\right) = R_{12} dt \tag{2.205b}$$
$$\mathrm{E}\left(de\, de^T\right) = R_2 dt. \tag{2.205c}$$

The values of the state variables and the outputs of the state-space system at discrete times kT_s, $k = 1, 2, \ldots$, are related through the stochastic difference equations

$$x(T_s k + T_s) = \Phi x(T_s k) + \bar{v}(T_s k) \tag{2.206a}$$
$$z(T_s k + T_s) = y(T_s k + T_s) - y(T_s k) = \theta x(T_s k) + \bar{e}(T_s k) \tag{2.206b}$$

where

$$\Phi = e^{AT_s} \tag{2.207a}$$
$$\theta = C \int_0^{T_s} e^{A\tau} d\tau \tag{2.207b}$$

and the discrete stochastic variables $\bar{v}(t)$ and $\bar{e}(t)$ have zero mean values and the covariances

$$\mathrm{E}\left(\bar{v}(t)\bar{v}^T(t)\right) = \bar{R}_1 = \int_0^{T_s} e^{A(T_s-\tau)} R_1 \left(e^{A(T_s-\tau)}\right)^T d\tau \tag{2.208a}$$

$$\mathrm{E}\left(\bar{v}(t)\bar{e}^T(t)\right) = \bar{R}_{12} = \int_0^{T_s} e^{A(T_s-\tau)} \left(R_1 \Theta^T(\tau) + R_{12}\right) d\tau \tag{2.208b}$$

$$\mathrm{E}\left(\bar{e}(t)\bar{e}^T(t)\right) = \bar{R}_2 = \int_0^{T_s} \left(\Theta(\tau) R_1 \Theta^T(\tau) + \Theta(\tau) R_{12} + R_{12}^T \Theta^T(\tau) + R_2\right) d\tau \tag{2.208c}$$

$$\Theta(\tau) = C \int_\tau^{T_s} e^{A(s-\tau)} ds. \tag{2.208d}$$

Proof: See Åström (1970). □

2.7.4 Nonlinear Stochastic Models

For nonlinear stochastic models, we will limit the discussion to the case when white noise $v(t)$ enters affinely into the equations,

$$\dot{x}(t) = f\big(x(t), t\big) + \sigma\big(x(t), t\big)v(t). \tag{2.209}$$

That the noise enters affinely into the equations is of course a special case of a more general model structure where the noise enters through a general nonlinear function. However, the general case is less treated in the literature. Since our goal is to extend existing results for state-space models to DAE models, the discussion is limited to the special case (2.209).

As in the linear case, (2.209) must be treated as a stochastic integral (Åström, 1970) To point this out, the notation

$$dx = f\big(x(t), t\big)dt + \sigma\big(x(t), t\big)\, dv, \tag{2.210}$$

where $v(t)$ is a Wiener process, is used.

Stochastic state models can be used for example for simulation and for state estimation using nonlinear filtering methods such extended Kalman filters (e.g., Kailath et al., 2000; Gustafsson, 2000) and particle filters (e.g., Gordon et al., 1993; Doucet et al., 2001; Ristic et al., 2004). Particle filters are also discussed in Chapter 4 of this thesis.

2.8 Conclusions

We introduced the concept of component-based modeling, and saw that this in the general case leads to a differential-algebraic equation (DAE). We discussed general theory about DAE models, including the analysis method by Kunkel and Mehrmann (2001). This theory shows that provided the DAE satisfies Property 2.1, one part of the variables is determined by state-space equations, and one part of the variables is determined by algebraic equations. One part of the variables may also be undetermined. If no variables are undetermined, the DAE is called regular. We also discussed how large DAE systems can be solved in practice.

For linear DAE systems, we presented the concept of regularity and noted that it is equivalent to the existence of a unique solution. We also discussed a canonical form that is well-known in the literature, and provided a proof that will allow numerical computation as will be discussed in Chapter 11. This canonical form was then used to derive a state-space description. To get this state-space description, the input may have to be redefined as one of its derivatives in the continuous-time case or future values in the discrete-time case. For the continuous-time case, the state-space description was then used to sample the system.

We also discussed stochastic models, and properties of stochastic models that will be needed in the thesis.

3

System Identification

The main topic of this thesis is estimation of unknown parameters in differential-algebraic equation models. This is an application of system identification, so in this chapter basic properties of system identification are discussed. The different methods are only discussed briefly. For a more thorough discussion, the reader is referred to, e.g., Ljung (1999).

3.1 Prediction Error Methods

System identification is about estimating models from measured input data and output data. The measured data set is denoted Z^N,

$$Z^N = \{u(t_0), y(t_0), ..., u(t_N), y(t_N)\} \tag{3.1}$$

where u are inputs to the system and y outputs from the system. To estimate models, we use equations that are parameterized using parameters θ. We thus have a model structure that is known apart from the values of the constant parameters θ. In this thesis we are concerned with *gray-box models* where the parameters θ have a physical interpretation (Bohlin, 1991; Graebe, 1990; Ljung, 1999).

These unknown parameters are selected such that the measured data and the solution of the equations fit as closely as possible. This produces an estimate $\hat{\theta}$. A standard way to compare the measured output with the solution of the equations, is to consider the model's prediction of the output at each time point, given the data up to but not including that time point. This leads to the prediction error method. We are thus interested in the one-step-ahead predictor,

$$\hat{y}(t_k|t_{k-1}, \theta). \tag{3.2}$$

This is the prediction of $y(t_k)$ given Z^{k-1} (and $u(t_k)$, $u(t_{k+1})$, $\ldots$, if it is necessary) using the model corresponding to the parameter value θ. The *prediction errors* $\varepsilon(t_k, \theta)$ are

then the difference between the predicted outputs $\hat{y}(t_k|t_{k-1},\theta)$ and the measured outputs $y(t_k)$:

$$\varepsilon(t_k,\theta) = y(t_k) - \hat{y}(t_k|t_{k-1},\theta) \tag{3.3}$$

The parameters are estimated by minimizing a norm of the prediction errors. One common choice is the quadratic criterion

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\varepsilon^T(t_k,\theta)\Lambda^{-1}\varepsilon(t_k,\theta) \tag{3.4}$$

for some positive definite matrix Λ that is chosen according to the relative importance of the components of $\varepsilon(t_k,\theta)$. The parameter estimate $\hat{\theta}$ is then computed as

$$\hat{\theta} = \arg\min_{\theta} V_N(\theta, Z^N). \tag{3.5}$$

The minimization is typically performed by a numerical search method, for example Gauss-Newton (Ljung, 1999).

Depending on the model structure, the predictor $\hat{y}(t_k|t_{k-1},\theta)$ is computed in different ways. Below we will list the cases that are of interest in this thesis.

- For linear discrete-time state-space models,

$$x(t_{k+1}) = A(t_k)x(t_k) + B(t_k)u(t_k) + K(t_k)v_1(t_k) \tag{3.6a}$$
$$y(t_k) = C(t_k)x(t_k) + v_2(t_k) \tag{3.6b}$$

  where v_1 and v_2 are white noise processes, the predictor is computed using the Kalman filter (e.g., Kalman, 1960; Anderson and Moore, 1979; Kailath et al., 2000).

- For linear continuous-time state-space models with discrete-time measurements,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + K(t)v_1(t) \tag{3.7a}$$
$$y(t_k) = C(t_k)x(t_k) + v_2(t_k) \tag{3.7b}$$

  where v_1 and v_2 are white noise processes, the predictor is computed using the Kalman filter for continuous-time systems with discrete-time measurements (e.g., Jazwinski, 1970).

- For nonlinear state-space systems with only measurement noise (output-error models)

$$\dot{x}(t) = f\big(x(t), u(t), \theta\big) \tag{3.8a}$$
$$y(t_k) = h\big(x(t_k), \theta\big) + e(t_k). \tag{3.8b}$$

  the predictor is computed by simulating the system.

- For nonlinear state-space systems with a more general noise model,

$$\dot{x}(t) = f\big(x(t), u(t), \theta\big) + \sigma\big(x(t), u(t), \theta\big)w(t) \tag{3.9a}$$
$$y(t_k) = h\big(x(t_k), \theta\big) + e(t_k). \tag{3.9b}$$

the solution of the prediction problem is a infinite-dimensional nonlinear filter (e.g., Wong and Hajek, 1985, Chapter 5). However, there are approximate methods such as extended Kalman filters (e.g., Gustafsson, 2000) and particle filters (Gordon et al., 1993; Doucet et al., 2001; Ristic et al., 2004). Particle filters are further discussed in Chapter 4.

For all these cases, the initial condition $x(t_0)$ must be considered as known or estimated along with the parameters θ.

3.2 The Maximum Likelihood Method

The maximum likelihood method estimates the unknown parameters by maximizing the probability of the measured output with respect to the unknown parameters. Given that the measured signals have the probability density function $f_y(\theta, Z^N)$, the parameters are estimated as

$$\hat{\theta} = \arg\max_{\theta} f_y(\theta, Z^N). \tag{3.10}$$

Maximizing the likelihood function is equivalent to maximizing the log-likelihood function $\log f_y(\theta, Z^N)$. The parameters are then estimated as

$$\hat{\theta} = \arg\max_{\theta} \log f_y(\theta, Z^N). \tag{3.11}$$

The likelihood function can be computed using the likelihood function of the predictors (Ljung, 1999, Lemma 5.1). For linear state-space systems with Gaussian noise processes, the likelihood function can also be computed directly using the Kalman filter. For nonlinear state-space models, the likelihood function can for example be approximated using the particle filter. This is discussed, e.g., by Andrieu et al. (2004).

3.3 Frequency Domain Identification

Frequency domain methods aim to estimate the unknown parameters θ from frequency domain data,

$$Z^N = \{U(\omega_1), Y(\omega_1), \ldots, U(\omega_N), Y(\omega_N)\}, \tag{3.12}$$

where $Y(\omega_k)$ and $U(\omega_k)$ are the discrete Fourier transforms of the corresponding time domain signals in the discrete-time case, or approximations of the Fourier transforms in the continuous-time case. References on frequency domain identification are e.g., Ljung (1999, Section 7.7) and Pintelon and Schoukens (2001). The $Y(\omega_k)$ and $U(\omega_k)$ can be obtained directly from the system using a measurement device providing frequency domain data, or calculated from time domain data. Here we will consider frequency domain identification for linear systems described by transfer functions,

$$y(t) = G(p, \theta)u(t) + H(p, \theta)e(t) \tag{3.13}$$

in the continuous-time case and

$$y(t_k) = G(q, \theta)u(t_k) + H(q, \theta)e(t_k) \tag{3.14}$$

in the discrete-time case where $H(\cdot,\theta)$ is assumed to have a causal inverse. Here p is the differentiation operator, $px(t) = \frac{d}{dt}x(t)$ and q is the time shift operator, $qx(t_k) = x(t_{k+1})$.

In order to estimate the parameters, a criterion like

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} |Y(\omega_k) - G(e^{i\omega_k}, \theta)U(\omega_k)|^2 W_k \tag{3.15}$$

in the discrete-time case, and

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} |Y(\omega_k) - G(i\omega_k, \theta)U(\omega_k)|^2 W_k \tag{3.16}$$

in the continuous-time case is minimized. The *weighting functions* W_k can be selected using the noise model $H(\cdot,\theta)$. If the noise model depends on θ, a second term usually has to be added to the criterion to get consistent estimates, see the book by Ljung (1999, Equation 7.147).

It should be noted that an advantage with frequency domain identification methods is that continuous-time models and discrete-time models can be handled in similar ways.

3.4 Identifiability

Identifiability of a model structure means that it is possible to define its parameters uniquely. Essentially it requires that different parameter values θ_1 and θ_2 ($\theta_1 \neq \theta_2$) give different model outputs. References on identifiability are, e.g., Walter (1982) and Ljung (1999, Chapter 4).

When discussing identifiability, we will limit the discussion to a deterministic DAE model with a vector of unknown parameters θ,

$$G\big(\dot{x}(t), x(t), \theta, u(t), t\big) = 0 \tag{3.17a}$$

$$y(t) = h\big(x(t)\big) \tag{3.17b}$$

where $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$. As before, the parameters $\theta \in \mathbb{R}^{n_\theta}$ range over the set $D_{\mathcal{M}} \subseteq \mathbb{R}^{n_\theta}$. Formally the following definitions of identifiability will be used.

Definition 3.1 (Global identifiability). The model (3.17) is *globally identifiable* at θ_0, x_0 for the input $u(t)$ if

$$\left.\begin{array}{c} \theta \in D_{\mathcal{M}} \\ y(\theta_0, t) = y(\theta, t) \text{ for all } t \end{array}\right\} \Rightarrow \theta = \theta_0 \tag{3.18}$$

where $y(\bar{\theta}, t)$ is the output y of (3.17) with the input $u(t)$, $\theta = \bar{\theta}$, and the consistent initial condition x_0. The system is globally identifiable if it is globally identifiable at all $\theta_0 \in D_{\mathcal{M}}$ and consistent initial conditions x_0.

Another interesting property is local identifiability.

Definition 3.2 (Local identifiability). The model (3.17) is *locally identifiable* at θ_0, x_0 for the input $u(t)$ if there exists a neighborhood V of θ_0 for which

$$\left.\begin{array}{c}\theta \in V \\ y(\theta_0, t) = y(\theta, t) \text{ for all } t\end{array}\right\} \Rightarrow \theta = \theta_0 \tag{3.19}$$

where $y(\bar{\theta}, t)$ is the output y of (3.17) with the input $u(t)$, $\theta = \bar{\theta}$, and the consistent initial condition x_0. The system is locally identifiable if it is locally identifiable at all $\theta_0 \in D_{\mathcal{M}}$ and consistent initial conditions x_0.

Differential Algebra

If a DAE (3.17) consists only of polynomial equations, it is possible to use differential algebra (Ritt, 1966) to examine identifiability as described by Ljung and Glad (1994). The idea of this method is to transform the equations of the original DAE into a new set of equations, where it is easy to see whether the model is identifiable or not.

The following result by Ljung and Glad (1994) shows how this can be done. Assume that a model structure is specified by (3.17) where the equations are polynomials and that the unknown parameters are time-invariant, i.e., the equations

$$\dot{\theta}(t) = 0 \tag{3.20}$$

are included among the equations (3.17). Using Ritt's algorithm from differential algebra (Ritt, 1966; Ljung and Glad, 1994) it is typically possible to compute a new set of polynomial equations of the form

$$\begin{aligned} A_1(y, u, p) &= 0 \\ &\vdots \\ A_{n_y}(y, u, p) &= 0 \\ B_1(y, u, \theta_1, p) &= 0 \\ B_2(y, u, \theta_1, \theta_2, p) &= 0 \\ &\vdots \\ B_{n_\theta}(y, u, \theta_1, \theta_2, \ldots, \theta_{n_\theta}, p) &= 0 \\ C_1(y, u, \theta, x, p) &= 0 \\ &\vdots \\ C_{n_x}(y, u, \theta, x, p) &= 0 \end{aligned} \tag{3.21}$$

which has the same solutions as (3.17) if some conditions of the form

$$s_i\big(x(t), y(t), u(t), \theta(t), p\big) \neq 0 \qquad i = 1, 2, \ldots, n_s \tag{3.22}$$

are satisfied. $\theta_1, \theta_2, \ldots, \theta_{n_\theta}$ are the scalar elements of the vector θ. In this set of equations, p is the differentiation operator $px(t) = \frac{d}{dt}x(t)$. For example, the expression $A_1(y, u, p)$ could also be written as $A_1(y, \dot{y}, \ddot{y}, \ldots, u, \dot{u}, \ddot{u}, \ldots)$. Also note that the requirement that

the original DAE only consists of polynomial equations is essential for Ritt's algorithm to be applicable.

Ritt's algorithm is a procedure that resembles Gram-Schmidt orthogonalization or Gauss elimination. For example, to produce $A_1(y, u, p)$ in (3.21), it takes an arbitrary equation of the original DAE. If this element contains unwanted features (e.g., θ occurs in the equation), they are removed with allowed algebraic manipulations (addition or multiplication of another element or its derivative). By this procedure, a "better" element is produced in each step, and after a finite number of steps it will find the desired form. See further Ritt (1966); Ljung and Glad (1994).

Since (3.21) is equivalent to (3.17), it is possible to use those equations to examine identifiability of (3.17). To do this we observe that only the B_i polynomials give information about the value of θ since the A_i polynomials do not include θ and the C_i include the x variables. Identifiability is thus determined by the polynomials B_i in (3.21). If the variables θ_1, θ_2,... all occur exclusively in undifferentiated form in the B_i (i.e., no terms $\dot{\theta}$ occur), then these polynomials give a triangular set of nonlinear equations for determining the θ_i. There are three cases that can occur, depending on the identifiability properties of the model.

1. If the B_i have the form

$$B_i = P_i(y, u, p)\theta_i - Q_i(y, u, p), \tag{3.23}$$

   i.e., a linear regression, then the model structure is globally identifiable, provided $P_i(y, u, p) \neq 0$.

2. If the B_i are higher order polynomials in θ, then there is local but not global identifiability.

3. If there are equations on the form

$$B_i = \dot{\theta}_i, \tag{3.24}$$

   then these θ_i are neither locally nor globally identifiable.

The method discussed in this section is illustrated in the following example.

Example 3.1: Identifiability and differential algebra

Consider the model structure

$$\ddot{y} + 2\theta\dot{y} + \theta^2 y = 0 \tag{3.25a}$$

$$\dot{\theta} = 0 \tag{3.25b}$$

from Ljung and Glad (1994). Applying Ritt's algorithm returns

$$A_1(y, p) = 4y\ddot{y}^3 - 3\dot{y}^2 \cdot \ddot{y}^2 - 6yy^{(3)}\dot{y}\ddot{y} + 4y^{(3)}\dot{y}^3 + y^2 y^{(3)2} \tag{3.26a}$$

$$B_1(y, \theta, p) = (\dot{y}\ddot{y} - yy^{(3)}) + 2\theta(\dot{y}^2 - y\ddot{y}). \tag{3.26b}$$

Since B_1 is a linear function in θ, the model structure is globally identifiable.

3.5 Observability

For the special case of DAE models,

$$G\big(\dot{x}(t), x(t), u(t), t\big) = 0 \tag{3.27a}$$

$$y(t) = h\big(x(t)\big), \tag{3.27b}$$

observability means that it is possible to uniquely estimate the internal variables $x(t)$ if the input $u(t)$ and output $y(t)$ are known. The internal variables may implicitly depend on derivatives of $u(t)$, so it will be assumed that the input is infinitely differentiable.

Observability is closely related to identifiability. The conceptual connections between observability and identifiability for nonlinear systems can be seen by noting that identifiability of θ in the DAE model

$$G\big(\dot{x}(t), x(t), \theta, u(t), t\big) = 0 \tag{3.28a}$$

$$y(t) = h\big(x(t)\big) \tag{3.28b}$$

can bee seen as observability of the (constant) variable $\theta(t)$ in the model

$$G\big(\dot{x}(t), x(t), \theta(t), u(t), t\big) = 0 \tag{3.29a}$$

$$\dot{\theta}(t) = 0 \tag{3.29b}$$

$$y(t) = h\big(x(t)\big). \tag{3.29c}$$

Observability is treated in books on nonlinear control systems such as the one by Nijmeijer and van der Schaft (1990) and the one by Isidori (1989). See also the article by Hermann and Krener (1977).

We will only discuss local weak observability, which means that the observability property only is examined in a region around the true trajectory of the internal variables. Formally, the following definitions are used. First, let the solution $x(t)$ of the DAE (3.27a) with the consistent initial condition x_0 and the input $u(t)$ be denoted $\pi\big(t; x_0, u(t)\big)$. Two consistent initial conditions x_1 and x_2 are then indistinguishable if they give rise to the same output, i.e.,

$$h\Big(\pi\big(t; x_1, u(t)\big)\Big) = h\Big(\pi\big(t; x_2, u(t)\big)\Big) \tag{3.30}$$

for all infinitely differentiable $u(t)$. A natural definition of observability is then that if x_1 and x_2 are indistinguishable, then $x_1 = x_2$. For local weak observability, which is the case that mainly is discussed in this thesis, a more involved definition is necessary.

Definition 3.3. Let U be an open set. Two consistent initial conditions x_1 and x_2 which both belong to U are said to be *U-indistinguishable* if they give the same outputs in all cases where both trajectories lie entirely in U, i.e.,

$$h\Big(\pi\big(t; x_1, u(t)\big)\Big) = h\Big(\pi\big(t; x_2, u(t)\big)\Big) \text{ for all } t \in [t_0, t_1] \tag{3.31}$$

as soon as

$$\pi\big(t; x_1, u(t)\big) \in U,\ \pi\big(t; x_2, u(t)\big) \in U,\ t \in [t_0, t_1] \tag{3.32}$$

for all infinitely differentiable inputs $u(t)$. The set of all points that are U-indistinguishable from x_0 is denoted $I_U(x_0)$.

It is now possible to give the definition of local weak observability:

Definition 3.4 (Local weak observability). The system (3.27) is *locally weakly observable* at the consistent initial condition x_0 if there exists an open neighborhood U of x_0 such that for every neighborhood V of x_0 with $V \subset U$, $I_V(x_0) = \{x_0\}$. If this is true for all points x_0, the system is locally weakly observable.

Part I

Nonlinear DAE Models

4

Well-Posedness of Nonlinear Estimation Problems

This chapter discusses noise modeling and estimation for nonlinear DAE systems. We will pay special attention to well-posedness of noise models.

4.1 Introduction

When modeling physical systems, it is usually impossible to predict the exact behavior of the system. This can have several explanations. One common situation is that it is known that external stimuli are affecting the systems, but these signals cannot be measured or chosen.

Example 4.1: Noise modeling: process noise

Consider an airplane with mass m flying straight ahead with velocity $x(t)$. The force produced by the engine is called $u(t)$. The resistance caused by the air if there were no wind, is the known function $f(x)$. If there is no wind, the motion of the aircraft is described by

$$m\dot{x}(t) = u(t) - f(x). \tag{4.1}$$

If there is wind acting on the aircraft, this can be seen as an additional force $w(t)$ acting on the aircraft. The motion of the aircraft is then described by

$$m\dot{x}(t) = u(t) - f(x) + w(t). \tag{4.2}$$

This force $w(t)$ is an example of an external stimulus that is known to exist, but cannot be measured.

As discussed in Section 2.1.2 and 2.7, external stimuli that are not measured are often

modeled as stochastic processes.

Another common situation is that certain signals in the system are measured, but there are imperfections in the measurements. For example, a sensor may have an unknown offset or produce measurements with a time-varying error.

Example 4.2: Noise modeling: measurement noise

Consider the airplane discussed in Example 4.1. Assume that a sensor is measuring the velocity $x(t)$ of the aircraft at time instances t_k, $k = 1, \ldots$ Due to imperfections in the measurement device, an unknown error $e(t_k)$ is added to each measurement $y(t_k)$. The measurement equation is then

$$y(t_k) = x(t_k) + e(t_k), \quad k = 1, \ldots \tag{4.3}$$

Also measurement noise can be modeled as a stochastic process. A third possibility is that a model has imperfections, but that these cannot be classified as unmeasured external stimuli or measurement imperfections.

Example 4.3: Noise modeling: model imperfections

Consider a system which is described using a state-space model,

$$\dot{x}(t) = f\big(x(t)\big). \tag{4.4}$$

Assume that observations show that (4.4) is only approximately satisfied. It may then be appropriate to include a term $w(t)$ to indicate that the equation does not hold exactly,

$$\dot{x}(t) = f\big(x(t)\big) + w(t). \tag{4.5}$$

Also in this case, the properties of $w(t)$ can be modeled as a stochastic process. We have now identified three situations when it may be appropriate to include stochastic processes when modeling a physical system:

- Unmeasured external stimuli are affecting the system.
- A signal is measured with an imperfect sensor.
- There are model imperfections that do not fall into the two previous categories.

As discussed in the examples above, the first and last case can often be modeled by including a stochastic process $w(t)$ in the model. This is called process noise. The second case is typically modeled by including a stochastic process $e(t_k)$ in the equations describing the measurements.

This chapter is about how we can handle the situations discussed above for DAE models. We are thus interested in incorporating process noise $w(t)$ and measurement noise $e(t_k)$ in a DAE model. In the general case, this would result in the stochastic DAE

$$F\big(\dot{x}(t), x(t), w(t), u(t)\big) = 0 \tag{4.6a}$$

$$y(t_k) = h\big(x(t_k)\big) + e(t_k) \tag{4.6b}$$

where u is a known input and y is a measured output. If there is no process noise, the stochastic DAE simplifies to

$$F\big(\dot{x}(t), x(t), u(t)\big) = 0 \tag{4.7a}$$

$$y(t_k) = h\big(x(t_k)\big) + e(t_k). \tag{4.7b}$$

This is called an output-error model. Once we have established how noise can be added to DAE models, we will also discuss how the internal variables can be estimated using particle filters (Gordon et al., 1993; Doucet et al., 2001; Ristic et al., 2004) and also how unknown parameters can be estimated.

4.2 Literature Overview

The question whether the state estimation problem for DAE models is well-defined has been discussed by, e.g., Schein and Denk (1998), Winkler (2004), Darouach et al. (1997), Kučera (1986), Germani et al. (2002), and Becerra et al. (2001). In Schein and Denk (1998), linear SDAE are treated, and it is guaranteed that the noise is not differentiated by assuming that the system has differential index 1 (see Section 2.2). The assumption that the system has differential index 1 is more restrictive than is necessary, and rules out some applications such as many mechanics systems. This assumption will not be made here. Schein and Denk (1998) also note that some internal variables actually may be so-called *generalized stochastic processes*, that is a time-continuous white noise process. Winkler (2004) makes the same assumption as Schein and Denk (1998), but also treats a class of nonlinear DAEs.

Darouach et al. (1997) treat linear DAEs with differential index 1, and a Kalman filter is constructed. However, in the estimation procedure the authors seem to overlook the fact that some variables may have infinite variance. In Kučera (1986), the original linear SDAE system specification may actually specify derivatives of white noise, but a controller is designed that removes any derivatives. In Germani et al. (2002) restrictive assumptions are made that guarantee that no derivatives appear in the linear SDAE, although this is not stated explicitly. Finally, in Becerra et al. (2001) nonlinear semi-explicit DAEs (e.g., Brenan et al., 1996) are discussed. Here well-posedness is guaranteed by only adding noise to the state-space part of the system.

4.3 Background and Motivation

As mentioned in the introduction, the question treated in this chapter is how unknown disturbances can be modeled in DAEs according to

$$F\big(\dot{x}(t), x(t), w(t), u(t)\big) = 0 \tag{4.8a}$$

$$y(t_k) = h\big(x(t_k)\big) + e(t_k), \tag{4.8b}$$

where w is process noise and e is measurement noise, and also how such models can be used for estimation of the internal variables $x(t)$ and constant parameters. We will limit

the discussion to the case when $w(t)$ is a Gaussian second order stationary process with spectrum

$$\phi_w(\omega). \tag{4.9}$$

The spectrum is assumed to be rational in ω with pole excess $2p_w$. This means that

$$\lim_{\omega\to\infty} \omega^{2p_w}\phi_w(\omega,\theta) = C(\theta)$$
$$0 < C(\theta) < \infty \text{ for } \theta \in D_{\mathcal{M}}.$$

An important property of DAE models is that the internal variables may depend on derivatives of the inputs to the model. This can for example be realized from the discussion on linear DAE models in Section 2.3. This is one of the central points when discussing noise for DAE models. Since $w(t)$ occurs as an input signal in the DAE equations (4.8), one or more of its derivatives with respect to time may affect the internal variables $x(t)$. This is a problem, since time derivatives of a Gaussian second order stationary process may not have finite variance. Actually, $w(t)$ can be differentiated at most $p_w - 1$ times since it has pole excess $2p_w$. This can be realized from (2.179b) which gives that the variance of $\frac{d^n}{dt^n}w(t)$ is

$$r(0) = \int_{-\infty}^{\infty} (iw)^{2n}\phi_w d\omega \begin{cases} < \infty & \text{if } n \le p_w - 1 \\ = \infty & \text{if } n \ge p_w. \end{cases} \tag{4.10}$$

Example 4.4: Noise modeling difficulties

Consider the DAE

$$\begin{pmatrix} \dot{x}_1(t) - x_2(t) \\ \dot{x}_3(t) - x_2(t) \\ x_1^2(t) + x_3^2(t) - 1 - w(t) \end{pmatrix} = 0 \tag{4.11}$$

where a stochastic process has been added to the last equation to model an unmeasured disturbance. Differentiating the last equation with respect to time gives

$$2x_1(t)\dot{x}_1(t) + 2x_3(t)\dot{x}_3(t) - \dot{w}(t) = 0. \tag{4.12}$$

Eliminating $\dot{x}_1(t)$ and $\dot{x}_3(t)$ using the first two equations of the DAE and solving for $x_2(t)$ gives

$$x_2(t) = \frac{\dot{w}(t)}{2x_1(t) + 2x_3(t)}. \tag{4.13}$$

If the spectrum of $w(t)$ has pole excess 2, this is questionable since $\dot{w}(t)$ then has infinite variance. However, if the pole excess is 3 or higher, the involved signals have finite variance.

As we saw in the example above, it is essential to examine how many derivatives of $w(t)$

that affect the internal variables. A central tool in this aspect will be the methods by Kunkel and Mehrmann (2001) that were reviewed in Section 2.2. The result that will be used is that given that a DAE fulfills Property 2.1, it is possible to see it as a combination of a state-space system that determines part of the variables, and algebraic equations that determine part of the variables. More specifically, consider the DAE

$$F\big(\dot{x}(t), x(t), w(t), u(t)\big) = 0 \tag{4.14}$$

where we for a moment assume that all involved signals can be differentiated as many times as necessary. If this DAE fulfills Property 2.1, then there exists matrices Z_1 and Z_2, and a constant integer μ such that

$$\hat{F}_1(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, u, w) = Z_1^T F \tag{4.15a}$$

$$\hat{F}_2(x_1, x_2, x_3, u, \dot{u}, \ldots, u^{(\mu)}, w, \dot{w}, \ldots, w^{(\mu)}) = Z_2^T \begin{pmatrix} F \\ \frac{d}{dt}F \\ \vdots \\ \frac{d^\mu}{dt^\mu}F \end{pmatrix} \tag{4.15b}$$

where the notation $u^{(\mu)}$ is used for the μ:th time derivative of the signal u. From these equations it is then, according to the theory by Kunkel and Mehrmann (2001), possible to solve for x_3 in $\hat{F}_2 = 0$, and after using that equation to eliminate $\dot{x}_3$ and x_3 in $\hat{F}_1$, the equation $\hat{F}_1 = 0$ can be solved for $\dot{x}_1$.

If we now again let $w(t)$ be a stochastic process which has a spectrum with pole excess $2p_w$, then it can be differentiated at most $p_w - 1$ times. If it is differentiated p_w times or more, the resulting signal has infinite variance. This means that a sufficient condition for the signals x in the DAE to have finite variance is that no derivatives of w higher than $p_w - 1$ occur in $\hat{F}_2$ in (4.15b). Throughout this chapter, we will assume that the DAEs are regular, so that x_2 is of size zero. This discussion leads to the following result.

Result 4.1
Consider the SDAE

$$F\big(\dot{x}(t), x(t), w(t), u(t)\big) = 0 \tag{4.16}$$

where $w(t)$ is a Gaussian second order stationary process with spectrum $\phi_w(\omega)$ which is rational in ω with pole excess $2p_w$. Assume that the SDAE, with $w(t)$ considered as a differentiable signal, fulfills Property 2.1 and is regular. The signals $x(t)$ then have finite variance provided that $\hat{F}_2$ can be written as

$$\hat{F}_2 = \hat{F}_2(x_1, x_2, x_3, u, \dot{u}, \ldots, u^{(k)}, w, \dot{w}, \ldots, w^{(l)}) \tag{4.17}$$

where $l \leq p_w - 1$ and $\hat{F}_2$ is defined by (4.15b).

The above discussion shows how it can be examined if a noise process $w(t)$ is differentiated too many times so that the resulting equations include signals with infinite variance. However, we would also like to be able to discuss solutions to stochastic DAEs in terms of stochastic differential equations. Our approach to this will be to convert the SDAE to the state-space form (2.210) discussed in Section 2.7.4. Then methods for stochastic state-space systems can then be used to define the solution.

The methods discussed in Section 2.7.4 require the noise process to be white noise, but in this chapter we have so far only discussed noise $w(t)$ with finite variance. However, as $w(t)$ is assumed to be a Gaussian second order stationary process, it can be seen as white noise filtered through a linear filter (e.g., Section 2.7). The filter can for example be written in state-space form,

$$\dot{x}_w(t) = Ax_w(t) + Bv(t) \tag{4.18a}$$
$$w(t) = Cx_w(t) \tag{4.18b}$$

where $v(t)$ is white noise. Combining the SDAE (4.8a) and (4.18) gives

$$F\big(\dot{x}(t), x(t), Cx_w(t), u(t)\big) = 0 \tag{4.19a}$$
$$\dot{x}_w(t) = Ax_w(t) + Bv(t) \tag{4.19b}$$

This can be seen as a single SDAE,

$$G\big(\dot{z}(t), z(t), v(t), u(t)\big) = 0 \tag{4.20}$$

where $v(t)$ is white noise and

$$z(t) = \begin{pmatrix} x(t) \\ x_w(t) \end{pmatrix}. \tag{4.21}$$

When the SDAE contains white noise terms, additional restrictions apply. Not only is it not allowed to differentiate the white noise signal, but it must also be integrated in the affine form discussed in Section 2.7.4.

Example 4.5: White noise modeling difficulties

Consider the nonlinear DAE

$$\dot{x}_1(t) - x_2^2(t) = 0 \tag{4.22a}$$
$$x_2(t) - v(t) = 0 \tag{4.22b}$$

where $v(t)$ is white noise. The second equation states that $x_2(t)$ is equal to a time-continuous white noise process. Since such processes have infinite variance, this is questionable if $x_2(t)$ represents a physical quantity. The first equation states that

$$\dot{x}_1(t) = v^2(t) \tag{4.23}$$

which also is questionable since nonlinear operations on white noise cannot be handled in the framework of stochastic integrals as discussed in Section 2.7.4.

The main topics of this chapter concern how noise can be included in DAE models without introducing problems such as those discussed in the example and how particle filters and parameter estimation can be implemented for DAE models with white noise inputs.

4.4 Main Results

The main result of this chapter states conditions for when a SDAE with white process noise v,

$$F\big(\dot{x}(t), x(t), v(t), u(t)\big) = 0, \tag{4.24}$$

can be interpreted as a stochastic differential equation, and thus has a well-defined solution. As discussed above, such models typically arise from a modeling situation where disturbances have been modeled as second order processes $w(t)$. These $w(t)$ have then been modeled as white noise $v(t)$ filtered through a linear filter according to the following procedure:

1. Let the process noise $w(t)$ of an SDAE

$$G\big(\dot{z}(t), z(t), w(t), u(t)\big) = 0 \tag{4.25}$$

be modeled as white noise $v(t)$ passed through a linear filter,

$$\dot{z}_w(t) = Az_w(t) + Bv(t) \tag{4.26a}$$
$$w(t) = Cz_w(t). \tag{4.26b}$$

2. Combine this into one SDAE with white process noise,

$$F\big(\dot{x}(t), x(t), v(t), u(t)\big) = 0 \tag{4.27}$$

where

$$F = \begin{pmatrix} G\big(\dot{z}(t), z(t), Cz_w(t), u(t)\big) \\ \dot{z}_w(t) - Az_w(t) - Bv(t) \end{pmatrix} \tag{4.28}$$

and

$$x(t) = \begin{pmatrix} z(t) \\ z_w(t) \end{pmatrix}. \tag{4.29}$$

To simplify the notation, we will let the SDAE depend directly on time t instead of on the input $u(t)$,

$$F\big(\dot{x}(t), x(t), v(t), t\big) = 0. \tag{4.30}$$

To formulate the result, we will use the functions $\hat{F}_1$ and $\hat{F}_2$ from the theory by Kunkel and Mehrmann (2001). $\hat{F}_1$ and $\hat{F}_2$ were introduced in Section 2.2 and also discussed in Section 4.3 of this chapter. From Section 2.2 we also recall the notation

$$\hat{F}_{l;p} \tag{4.31}$$

for partial derivatives of $\hat{F}_l$ with respect to the variables p, e.g.,

$$\hat{F}_{2;x_1,x_2} = \begin{pmatrix} \frac{\partial}{\partial x_1}\hat{F}_2 & \frac{\partial}{\partial x_2}\hat{F}_2 \end{pmatrix}. \tag{4.32}$$

Furthermore, we will denote the inverse of the square matrix $\hat{F}_{2;x_3}$ by

$$\hat{F}_{2;x_3}^{-1}. \tag{4.33}$$

Theorem 4.1
Assume that (4.30) *satisfies Corollary 2.1 when $v(t)$ is considered as a known signal of which we can take formal derivatives. Let $\hat{F}_1$, $\hat{F}_2$, x_1, x_2, and x_3 be defined as in Section 2.2 and assume that the system is regular (x_2 is of size zero).*

Then there exists a well-defined solution x in terms of stochastic differential equations to (4.30) *with $v(t)$ considered as white noise provided that $\hat{F}_1$ and $\hat{F}_2$ can be written as*

$$\hat{F}_1 = \hat{F}_1\big(t, x_1, x_3, \dot{x}_1 - \sigma(x_1, x_3)v, \dot{x}_3 + \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v\big) \tag{4.34a}$$

$$\hat{F}_2 = \hat{F}_2\big(t, x_1, x_3\big) \tag{4.34b}$$

for some function $\sigma(x_1, x_3)$.

Proof: Differentiating (4.34b) with respect to time yields

$$\hat{F}_{2;t} + \hat{F}_{2;x_1}\dot{x}_1 + \hat{F}_{2;x_3}\dot{x}_3 = 0. \tag{4.35}$$

Since $\hat{F}_2$ is locally solvable for x_3, $F_{2;x_3}$ is invertible. This means that $\dot{x}_3$ can be written as

$$\dot{x}_3 = -\hat{F}_{2;x_3}^{-1}(\hat{F}_{2;t} + \hat{F}_{2;x_1}\dot{x}_1). \tag{4.36}$$

(4.34b) can also be locally solved for x_3 to give

$$x_3 = \mathcal{R}(t, x_1) \tag{4.37}$$

for some function $\mathcal{R}$. Inserting this into (4.34a) gives

$$\hat{F}_1\big(t, x_1, \mathcal{R}, \dot{x}_1 - \sigma(x_1, \mathcal{R})v, -\hat{F}_{2;x_3}^{-1}(\hat{F}_{2;t} + \hat{F}_{2;x_1}\dot{x}_1) + \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, \mathcal{R})v\big). \tag{4.38}$$

The equation $\hat{F}_1 = 0$ now takes the form

$$\hat{F}_1\big(t, x_1, \mathcal{R}, \dot{x}_1 - \sigma(x_1, \mathcal{R})v, -\hat{F}_{2;x_3}^{-1}\hat{F}_{2;t} - \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}(\dot{x}_1 - \sigma(x_1, \mathcal{R})v)\big) = 0. \tag{4.39}$$

Since Corollary 2.1 is fulfilled, this equation can be solved for $\dot{x}_1$. Since $-\sigma(x_1, \mathcal{R})v$ enters the equations in the same way as $\dot{x}_1$, the solution takes the form

$$\dot{x}_1 - \sigma(x_1, \mathcal{R})v = \mathcal{L}(t, x_1) \tag{4.40}$$

for some function $\mathcal{L}$. This can be interpreted as the stochastic differential equation

$$dx_1 = \mathcal{L}(t, x_1)dt + \sigma(x_1, \mathcal{R})dv \tag{4.41}$$

so x_1 has a well-defined solution. A solution for x_3 is then defined through (4.37). □

If noise has been added to a DAE model using physical insight or for other reasons, the theorem above gives conditions for the system to be well-posed using a transformed version of the system. It may also be interesting to be able to see if the SDAE is well-posed already in the original equations. As discussed in the theorem above, the SDAE is well-posed if the equations $\hat{F}_1 = 0$ and $\hat{F}_2 = 0$ take the form

$$\hat{F}_1\big(t, x_1, x_3, \dot{x}_1 - \sigma(x_1, x_3)v, \dot{x}_3 + \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v\big) = 0 \tag{4.42a}$$

$$\hat{F}_2\big(t, x_1, x_3\big) = 0. \tag{4.42b}$$

In the original equations, this can typically be seen as adding noise according to

$$F\left(\begin{pmatrix} \dot{x}_1 - \sigma(x_1, x_3)v \\ \dot{x}_3 + \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v \end{pmatrix}, \begin{pmatrix} x_1 \\ x_3 \end{pmatrix}, t\right) = 0. \tag{4.43}$$

One common situation when it is easy to see how white noise can be added is a semi-explicit DAE (Brenan et al., 1996) with differential index 1. This is considered in the following example.

Example 4.6: Noise modeling: semi-explicit index 1 DAE

Consider a semi-explicit DAE with differential index 1,

$$\dot{x}_a = f(x_a, x_b) \tag{4.44a}$$
$$0 = g(x_a, x_b). \tag{4.44b}$$

Locally, x_b can be solved from (4.44b), so these equations correspond to $\hat{F}_1 = 0$ and $\hat{F}_2 = 0$ respectively. Noise can thus be added according to

$$\dot{x}_a = f(x_a, x_b) + \sigma(x_a, x_b)v \tag{4.45a}$$
$$0 = g(x_a, x_b). \tag{4.45b}$$

4.5 Particle Filtering

An important aspect of uncertain models is state estimation and prediction. For nonlinear systems this is a difficult problem (e.g., Ristic et al., 2004; Andrieu et al., 2004; Schön, 2006). It is therefore necessary to resort to approximate methods. One approximate method for nonlinear state estimation is the *particle filter* (e.g., Gordon et al., 1993; Doucet et al., 2001; Ristic et al., 2004). In this section we will discuss how particle filter methods can be extended for use with SDAE models.

To be able to describe how existing particle filtering algorithms can be extended to DAE systems, we will first briefly describe how particle filtering can be implemented for state-space systems. Fore a more thorough treatment, see e.g., (Gordon et al., 1993; Doucet et al., 2001; Ristic et al., 2004). Existing particle filtering methods may allow other model structures than state-space systems, but we will limit the discussion here to state-space systems since that is enough to extend particle filtering methods to SDAE models.

Consider a nonlinear discrete-time state-space system,

$$x(t_{k+1}) = f\big(x(t_k), u(t_k), w(t_k)\big) \tag{4.46a}$$
$$y(t_k) = h\big(x(t_k)\big) + e(t_k) \tag{4.46b}$$

where x is the state vector, u is a known input, y is a measured output, and w and e are stochastic processes with known probability density functions. The particle filter is based on estimating the probability density function of the state $x(t_k)$, given the measurements

$$Z^N = \{u(t_0), y(t_0), ..., u(t_N), y(t_N)\}. \tag{4.47}$$

We are thus interested in computing the probability density function

$$p\big(x(t_k)|Z^N\big). \tag{4.48}$$

Depending on if $k < N$, $k = N$, or $k > N$ we will have a *smoothing problem*, *filtering problem*, and *prediction problem* respectively. Here we will limit the discussion to the filtering problem and the one-step-ahead prediction problem, that is we will have $N = k$ or $N = k - 1$.

Once (the estimate of) the probability density function has been computed, it can be used to estimate the value of $x(t)$. One possibility is to use the expected value of $x(t_k)$ given Z^N, another is to use the maximum a posteriori estimate, that is the $x(t_k)$ that maximizes $p\big(x(t_k)|Z^N\big)$.

In the particle filter, the probability density function (4.48), here with $N = k - 1$, is approximated by a sum of generalized Dirac functions,

$$p\big(x(t_k)|Z^{k-1}\big) \approx \sum_{i=1}^{M} q^{(i)}_{t_k|t_{k-1}} \delta\big(x(t_k) - x^{(i)}_{t_k|t_{k-1}}\big). \tag{4.49}$$

This means that the density function is approximated using M *particles*

$$\{x^{(i)}_{t_k|t_{k-1}}\}_{i=1}^{M} \tag{4.50}$$

with associated weights,

$$\{q^{(i)}_{t_k|t_{k-1}}\}_{i=1}^{M}. \tag{4.51}$$

Since the approximation is made using Dirac functions, it is not an approximation at each point x. Instead, the approximation holds for integrals of p. We can for example estimate the mean value of $x(t_k)$ as

$$\mathrm{E}\left(x(t_k)|Z^{k-1}\right) = \int x \cdot p\big(x(t_k)|Z^{k-1}\big)dx \approx \sum_{i=1}^{M} q^{(i)}_{t_k|t_{k-1}} x^{(i)}_{t_k|t_{k-1}}. \tag{4.52}$$

Now assume that a new measurement $\{y(t_k), u(t_k)\}$ is obtained. Using Bayes's rule, the probability density function $p\big(x(t_k)|Z^{k-1}\big)$ should be updated according to

$$p\big(x(t_k)|Z^{k}\big) = \frac{p\big(y(t_k)|x(t_k)\big)\, p\big(x(t_k)|Z^{k-1}\big)}{p\big(y(t_k)|Z^{k-1}\big)}. \tag{4.53}$$

Since $p\big(y(t_k)|Z^{k-1}\big)$ does not depend on x, the particle filter updates its approximation of the probability density function by updating the weights $\{q^{(i)}_{t_k|t_{k-1}}\}_{i=1}^{M}$ according to

$$q^{(i)}_{t_k|t_k} = \frac{p\big(y(t_k)|x^{(i)}_{t_k|t_{k-1}}\big) q^{(i)}_{t_k|t_{k-1}}}{\sum_{j=1}^{M} p\big(y(t_k)|x^{(j)}_{t_k|t_{k-1}}\big) q^{(j)}_{t_k|t_{k-1}}}, \quad i = 1, \ldots, M. \tag{4.54}$$

For the state space description (4.46), we have that

$$p\left(y(t_k)\,\middle|\, x^{(i)}_{t_k|t_{k-1}}\right) = p_e\left(y(t_k) - h(x^{(i)}_{t_k|t_{k-1}})\right) \tag{4.55}$$

where p_e is the probability density function of $e(t_k)$.

After this step, called the measurement update, the resampling step takes place. The resampling step redistributes the particles to avoid degeneration of the filter. It does not introduce additional information (actually, information is lost). We will use so-called *sampling importance resampling*. For other alternatives, see the references. The resampling step is in this case performed by replacing the M particles with M new particles. This is done by drawing M particles with replacement from the old particles. The probability to draw particle i is proportional to its weight $q^{(i)}_{t_k|t_k}$. The new particles $x^{(i)}_{t_k|t_k}$ are thus chosen according to

$$\Pr\left(x^{(i)}_{t_k|t_k} = x^{(j)}_{t_k|t_{k-1}}\right) = q^{(j)}_{t_k|t_k} \quad i = 1, \ldots, M. \tag{4.56}$$

The weights are changed to

$$q^{(i)}_{t_k|t_k} = \frac{1}{M} \quad i = 1, \ldots, M \tag{4.57}$$

so that the approximation of the probability density function is, approximately, left unchanged.

After the resampling step, the time update step takes place. This means that $x(t_{k+1})$ is predicted using available information about $x(t_k)$. For the particle filter and the state-space model (4.46), this is done by drawing M independent samples of $w(t_k)$, $w^{(i)}(t_k)$, $i = 1, \ldots, M$, according to its probability density function p_w. The particles are then updated according to

$$x^{(i)}_{t_{k+1}|t_k} = f\big(x^{(i)}_{t_k|t_k}, u(t_k), w^{(i)}(t_k)\big), \quad i = 1, \ldots, M. \tag{4.58}$$

In general, this can be seen as drawing new particles according to their conditional distribution,

$$x^{(i)}_{t_{k+1}|t_k} \sim p\left(x_{t_{k+1}|t_k} \middle| x^{(i)}_{t_k|t_k}\right), \quad i = 1, \ldots, M. \tag{4.59}$$

The weights are unchanged, $q^{(i)}_{t_{k+1}|t_k} = q^{(i)}_{t_k|t_k} = \frac{1}{M}$. Note that a more general version of the time update equation is available, see the references. After this step, a new measurement is obtained and the filter is restarted from the measurement update step.

When starting a filter, the particles should be initialized according to available information about the initial value, $x(t_0)$. If the probability density function of $x(t_0)$ is p_{x_0}, the particles are initially chosen according to that distribution. We can write this as

$$x^{(i)}_{t_0|t_{-1}} \sim p_{x_0}(x_0), \quad i = 1, \ldots, M \tag{4.60}$$

and we get

$$q^{(i)}_{t_0|t_{-1}} = \frac{1}{M}, \quad i = 1, \ldots, M. \tag{4.61}$$

Summing up, we get the following particle filtering algorithm.

1. Initialize the M particles,

$$x^{(i)}_{t_0|t_{-1}} \sim p_{x_0}(x_0), \quad i = 1, \ldots, M \tag{4.62}$$

and

$$q^{(i)}_{t_0|t_{-1}} = \frac{1}{M}, \quad i = 1, \dots, M. \tag{4.63}$$

Set $k := 0$.

2. Measurement update: calculate weights $\{q^{(i)}_{t_k|t_k}\}_{i=1}^M$ according to

$$q^{(i)}_{t_k|t_k} = \frac{p\big(y(t_k)|x^{(i)}_{t_k|t_{k-1}}\big)q^{(i)}_{t_k|t_{k-1}}}{\sum_{j=1}^M p\big(y(t_k)|x^{(j)}_{t_k|t_{k-1}}\big)q^{(j)}_{t_k|t_{k-1}}}, \quad i = 1, \dots, M. \tag{4.64}$$

3. Resampling: draw M particles, with replacement, according to

$$\Pr\left(x^{(i)}_{t_k|t_k} = x^{(j)}_{t_k|t_{k-1}}\right) = q^{(j)}_{t_k|t_k} \quad i = 1, \dots, M \tag{4.65}$$

and set

$$q^{(i)}_{t_{k+1}|t_k} = \frac{1}{M} \quad i = 1, \dots, M. \tag{4.66}$$

4. Time update: predict new particles according to

$$x^{(i)}_{t_{k+1}|t_k} \sim p\left(x_{t_{k+1}|t_k} \middle| x^{(i)}_{t_k|t_k}\right), \quad i = 1, \dots, M. \tag{4.67}$$

5. Set $k := k + 1$ and iterate from step 2.

To examine how the implementation for DAE systems should be done, we consider an SDAE in the form (4.6),

$$G\big(\dot{z}(t), z(t), w(t), u(t)\big) = 0 \tag{4.68a}$$

$$y(t_k) = h\big(z(t_k)\big) + e(t_k). \tag{4.68b}$$

To be able to use methods for stochastic simulation with white noise inputs, we realize the stochastic process $w(t)$ as white noise $v(t)$ filtered through a linear filter according to what was discussed in Section 4.3. Following the discussion in Section 4.3, we can then write the system as

$$F\big(\dot{x}(t), x(t), v(t), u(t)\big) = 0 \tag{4.69a}$$

$$y(t_k) = h\big(x(t_k)\big) + e(t_k). \tag{4.69b}$$

We only consider SDAE models (4.69a) that fulfill the conditions of Theorem 4.1. The theorem gives that we can write the system as

$$\hat{F}_1\big(u(t), x_1, x_3, \dot{x}_1 - \sigma(x_1, x_3)v, \dot{x}_3 + \hat{F}^{-1}_{2;x_3}\hat{F}_{2;x_1}\sigma(x_1, x_3)v\big) = 0 \tag{4.70a}$$

$$\hat{F}_2\big(u(t), \dot{u}(t), \dots, x_1, x_3\big) = 0 \tag{4.70b}$$

$$x(t) = Q\begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} \tag{4.70c}$$

$$y(t_k) = h\big(x(t_k)\big) + e(t_k). \tag{4.70d}$$

for some permutation matrix Q.

Since $\hat{F}_1$ and $\hat{F}_2$ are the result of the transformations discussed in Section 2.2, $\hat{F}_2$ can be locally solved for x_3,

$$x_3 = \mathcal{R}\big(u(t), \dot{u}(t), \ldots, x_1(t)\big). \tag{4.71}$$

After using (4.71) to eliminate x_3 and $\dot{x}_3$ in $\hat{F}_1$, $\hat{F}_1$ can be solved for $\dot{x}_1$ to give

$$\dot{x}_1 = \mathcal{L}(t, x_1) + \sigma(x_1, \mathcal{R})v. \tag{4.72}$$

Combining (4.70)–(4.72) gives

$$\dot{x}_1 = \mathcal{L}(t, x_1) + \sigma(x_1, \mathcal{R})v \tag{4.73a}$$

$$y(t_k) = h\left(Q\begin{pmatrix} x_1(t_k) \\ \mathcal{R}(u(t_k), \dot{u}(t_k), \ldots, x_1(t_k)) \end{pmatrix}\right) + e(t_k). \tag{4.73b}$$

The state-space system (4.73) can be used to implement a particle filter for estimation of x_1. After estimating x_1, estimates of x_3 can be computed using (4.71).

Since it is typically not possible to solve for $\dot{x}_1$ and x_3 explicitly, we will discuss numerical implementation methods in the following section. Furthermore, the state equation should be discretized. This can be done using for example a numerical solver for stochastic differential equations. The time update in step 4 in the particle filtering algorithm is thus performed by solving (4.73a) for one time step. The measurement update in step 2 of the particle filtering algorithm is performed using the measurement equation (4.73b).

4.6 Implementation Issues

The exact transformation into the form (4.70), which is necessary to implement the particle filter may be difficult to compute in practice. It is also an issue how to solve these equations numerically for $\dot{x}_1$ and x_3. Therefore approximate implementations may be considered. One way to do this is to use the type of DAE solver that is included in modeling environments for object-oriented modeling such as Dymola (Mattsson et al., 1998).

As discussed in Section 2.5, DAE solvers for component-based models compute an approximation of the form

$$\tilde{F}_1(t, x_1, x_3, \dot{x}_1) = 0 \tag{4.74a}$$

$$\hat{F}_2(t, x_1, x_3) = 0, \tag{4.74b}$$

that is $\hat{F}_1$ and $\hat{F}_2$ with $\dot{x}_3$ eliminated from $\hat{F}_1$. This can be used to examine if a DAE with a noise model satisfies the conditions of Theorem 4.1. The most straightforward way to check if a given noise model is correct, is to examine if the transformed system is of the form

$$\tilde{F}_1\big(t, x_1, x_3, \dot{x}_1 - \sigma(x_1, x_3)v\big) = 0 \tag{4.75a}$$

$$\hat{F}_2(t, x_1, x_3) = 0. \tag{4.75b}$$

If v appears in incorrect positions (so that the transformed system is not of the form (4.75)), one way to handle the situation would be to remove $v(t)$ from these incorrect locations in $\tilde{F}_1$ and $\hat{F}_2$, and assumed that noise is added to the original equations so that this is achieved.

The solvers can also be used for approximate implementation of particle filters for DAE systems. The idea behind this is that the transformation to the form

$$\dot{x}_1 = \mathcal{L}(t, x_1) + \sigma(x_1, \mathcal{R})v \tag{4.76a}$$

$$x_3 = \mathcal{R}(t, x_1) \tag{4.76b}$$

can be made by solving $\tilde{F}_1$ and $\hat{F}_2$ numerically at each time step using a DAE solver. This means that given values of x_1 and v the solver can give $\dot{x}_1$ and x_3. The state equation (4.76a) can then be used to estimate x_1, and x_3 can be computed from (4.76b).

To summarize, the following procedure can be used when modeling noise in DAEs and implementing a particle filter. First a DAE without noise is produced by writing down equations, or from component-based modeling. This DAE is then entered into a DAE solver to determine which variables that are states. Noise is then added to the original equations according to physical insight and then the equations are transformed into $\tilde{F}_1$ and $\hat{F}_2$. Then incorrect noise terms are removed so that the equations are in the form (4.75). The form (4.75) is then used to implement the particle filter by solving for $\dot{x}_1$ and x_3 using the DAE solver.

4.7 Example: Dymola Assisted Modeling and Particle Filtering

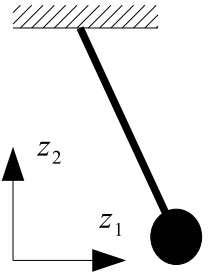


Figure 4.1: *A pendulum*

In this section we examine a DAE model of a pendulum. First noise is added, and then a particle filter is implemented to estimate the internal variables of the pendulum.

This is a modified example from Brenan et al. (1996). As shown in Figure 4.1, z_1 and z_2 are the horizontal and vertical position of the pendulum. Furthermore, z_3 and z_4 are the respective velocities, z_5 is the tension in the pendulum, the constant b represents resistance caused by the air, g is the gravity constant, and L is the constant length of the

pendulum. The equations describing the pendulum are

$$\dot{z}_1 = z_3 \tag{4.77a}$$
$$\dot{z}_2 = z_4 \tag{4.77b}$$
$$\dot{z}_3 = -z_5 \cdot z_1 - b \cdot z_3^2 \tag{4.77c}$$
$$\dot{z}_4 = -z_5 \cdot z_2 - b \cdot z_4^2 - g \tag{4.77d}$$
$$0 = z_1^2 + z_2^2 - L^2. \tag{4.77e}$$

We will use the approximate methods discussed in Section 4.6, so the equations are entered into the DAE solver in Dymola. The first step in the noise modeling is to let Dymola select which variables that are states. There are several possible ways to select states for these equations, but here z_1 and z_3 are selected. We thus have

$$x_1 = \begin{pmatrix} z_1 \\ z_3 \end{pmatrix}, \; x_3 = \begin{pmatrix} z_2 \\ z_4 \\ z_5 \end{pmatrix}. \tag{4.78}$$

We can thus take $\hat{F}_1$ as

$$\hat{F}_1 = \begin{pmatrix} \dot{z}_1 - z_3 \\ \dot{z}_3 - (-z_5 \cdot z_1 - b \cdot z_3^2) \end{pmatrix} \tag{4.79}$$

corresponding to (4.77a) and (4.77c). White noise could thus be added to the states z_1 and z_3. We choose to add noise only to $\dot{z}_3$ to model disturbances caused by e.g., turbulence. (4.77a) and (4.77c) then take the form

$$\dot{z}_1 = z_3 \tag{4.80a}$$
$$\dot{z}_3 = -z_5 \cdot z_1 - b \cdot z_3^2 + v \tag{4.80b}$$

where v is white noise. This corresponds to

$$\sigma = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.81}$$

in (4.34). The next step in the noise modeling is to transform these equations together with the remaining noise-free equations into $\tilde{F}_1$ and $\hat{F}_2$ in (4.75). Doing this reveals that $\tilde{F}_1$, which is available as C code from Dymola, is of the desired form

$$\tilde{F}_1\big(t, x_1, x_3, \dot{x}_1 - \sigma(x_1, x_2)v\big), \tag{4.82}$$

that is, the noise term only occurs in affine form and together with $\dot{x}_1$. However, $\hat{F}_2$ includes the noise term v which is not allowed. To solve this problem, occurrences of v in $\hat{F}_2$ are deleted before it is used for particle filtering. Removing the noise from F_2 can typically be seen as adding noise in the original equations, but a user does not need to consider the exact form of this. (For illustration, we will anyway discuss this below.)

Next, we implement a particle filter to estimate the internal variables of the system. To generate data for the estimation experiment, the model is inserted into the Simulink

environment using the Dymola-Simulink interface available with Dymola. The purpose of this experiment is not to demonstrate the performance of a filtering algorithm, but rather to show how DAE models can be used in a direct way when constructing particle filters. Therefore it is sufficient to use simulated data for the experiment. The constants were chosen as $L = 1$, $b = 0.05$ and $g = 9.81$. Process noise was generated with the *Band-Limited White Noise* block in Simulink with noise power 0.01. The initial values of the states were $z_1 = 0.5$ and $z_3 = -0.1$. The measured variable is the tension in the pendulum z_5,

$$y(t_k) = z_5(t_k) + e(t_k). \tag{4.83}$$

Measurements with noise variance 0.1 was collected at the sampling interval 0.05 s.

After generating the data, a particle filter was implemented using the algorithm in Section 4.5 to estimate the internal variables z_1, z_2, z_3, z_4, and z_5. Since the selected states are z_1 and z_3, these are the variables that are estimated directly by the particle filter. The remaining variables are then computed by Dymola using $\hat{F}_2$.

The particle filter was implemented in MATLAB with the time updates being performed by simulating the model using the Dymola-Simulink interface. The initial particles were spread between $z_1 = 0.1$ and $z_1 = 0.6$ and between $z_3 = -0.2$ and $z_3 = 0.2$. Only positive values of z_1 were used since the symmetry in the system makes it impossible to distinguish between positive and negative z_1 using only measurements of z_5. The particle filter was tuned to use noise power 0.1 for the process noise and variance 0.2 for the measurement noise to simulate the situation were the noise characteristics are not exactly known. A typical result of an estimation is shown in Figure 4.2 where an estimation of z_1 is plotted together with the true value.

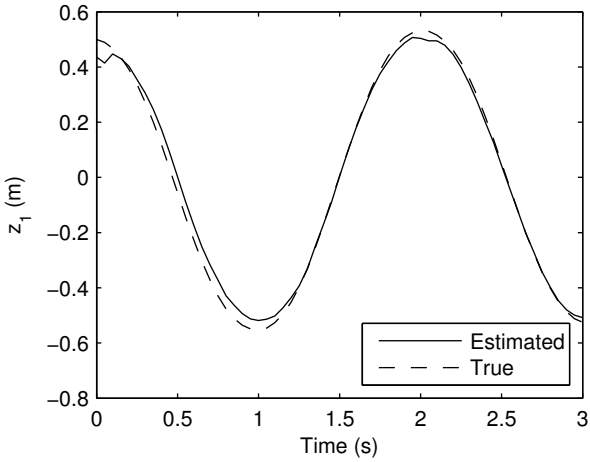


Figure 4.2: *Typical result of particle filtering.*

To examine the reliability of the filtering algorithm, 100 Monte Carlo runs were made.

Then the RMSE value was calculated according to

$$\mathrm{RMSE}(t) = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\big(x(t) - \hat{x}_j(t)\big)^2} \tag{4.84}$$

where M is the number of runs, here $M = 100$, $x(t)$ is the true state value and $\hat{x}_j(t)$ is the estimated state value in run j. The result is shown in Figure 4.3. The estimation error

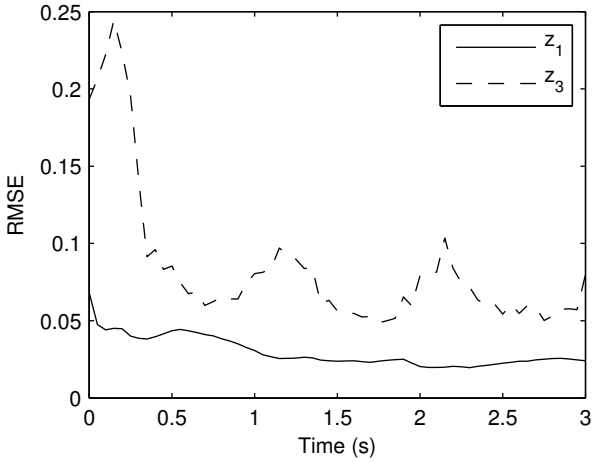


Figure 4.3: *RMSE for the estimations of z_1 and z_3 for 100 Monte Carlo runs.*

in the velocity z_3 is larger when the pendulum changes direction, which could mean that it is more difficult to estimate the velocity there.

Noise modeling details

When adding noise to a DAE, a user can only add noise so that it enters through a function σ. This was done in equation (4.80) above. However, noise must also be added according to the term $F_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v$ in (4.43) to make all variables well-defined (otherwise the conditions of Theorem 4.1 will not be satisifed).

To compute $\hat{F}_2$, we consider (4.77).

$$0 = z_1^2 + z_2^2 - L^2. \tag{4.85}$$

Differentiating (4.85) with respect to time gives

$$0 = 2z_1\dot{z}_1 + 2z_2\dot{z}_2. \tag{4.86}$$

Inserting (4.77a) and (4.77b) gives

$$0 = 2z_1z_3 + 2z_2z_4 \tag{4.87}$$

which after differentiation gives

$$0 = 2\dot{z}_1 z_3 + 2z_1\dot{z}_3 + 2\dot{z}_2 z_4 + 2z_2\dot{z}_4. \tag{4.88}$$

Inserting the expressions for the derivatives gives

$$0 = 2z_3^2 + 2z_1(-z_5 \cdot z_1 - bz_3^2) + 2z_4^2 + 2z_2(-z_5 \cdot z_2 - bz_4^2 - g). \tag{4.89}$$

The equations (4.85), (4.87), and (4.89) together define one possible selection of $\hat{F}_2$. These can be used to compute

$$F_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v = F_{2;x_3}^{-1}\hat{F}_{2;x_1}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\ \frac{z_1}{z_2}\\ *\end{pmatrix} \tag{4.90}$$

where the last term $*$ is unimportant since $\dot{z}_5$ does not occur in the equations. This tells us that noise should be added to $\dot{z}_4$ according to

$$\dot{z}_4 + \frac{z_1}{z_2}v = -z_5 \cdot z_2 - b \cdot z_4^2 - g \tag{4.91}$$

to satisfy the conditions of Theorem 4.1.

4.8 Parameter Estimation

In the parameter estimation problem, we are interested in estimating unknown parameters θ in a nonlinear SDAE with sampled measurements,

$$F\big(\dot{x}(t), x(t), u(t), v(t), \theta\big) = 0 \tag{4.92a}$$

$$y(t_k) = h(x(t_k), \theta) + e(t_k). \tag{4.92b}$$

As discussed in Section 4.3, the white noise $v(t)$ is typically the result of physical noise modeling, where low bandwidth noise is realized as white noise filtered through a linear filter. As in Chapter 3, we denote the measurements by

$$Z^N = \{u(t_0), y(t_0), ..., u(t_N), y(t_N)\}. \tag{4.93}$$

The parameters θ belong to the domain $D_{\mathcal{M}} \subseteq R^{n_\theta}$, $\theta \in D_{\mathcal{M}}$.

To guarantee that the model is well-defined, it will be assumed that it fulfills the conditions of Theorem 4.1 so that it can be written as

$$\hat{F}_1\big(u, \theta, x_1, x_3, \dot{x}_1 - \sigma(x_1, x_3)v, \dot{x}_3 + \hat{F}_{2;x_3}^{-1}\hat{F}_{2;x_1}\sigma(x_1, x_3)v\big) = 0 \tag{4.94a}$$

$$\hat{F}_2\big(u, \theta, x_1, x_3\big) = 0 \tag{4.94b}$$

$$y(t_k) = h\big(x_1(t_k), x_3(t_k), \theta\big) + e(t_k) \tag{4.94c}$$

for all $\theta \in D_{\mathcal{M}}$. As discussed previously in the chapter, this means that a particle filter can be implemented (for each value of θ). The particle filter gives two possibilities to

estimate the unknown parameters, the maximum likelihood method and the prediction error method (see Chapter 3).

The maximum likelihood method using particle filters is discussed by Andrieu et al. (2004). Here the probability density function of the output, $f_y(\theta, Z^N)$ is estimated using the particle filter. The parameters are thus estimated by maximizing the likelihood function,

$$\hat{\theta} = \arg\max_{\theta} f_y(\theta, Z^N) \tag{4.95}$$

or the log-likelihood function,

$$\hat{\theta} = \arg\max_{\theta} \log f_y(\theta, Z^N). \tag{4.96}$$

In can be noted that even though we have modeled all uncertainties as stochastic processes (4.92a), it is not straightforward to compute the likelihood function since there are several tuning parameters in the particle filter. For example, the number of particles and the resampling technique must be specified. Another issue, which is discussed by Andrieu et al. (2004), is that the particle filter itself is stochastic. This means that the estimate of the likelihood function $f_y(\theta, Z^N)$ will not be a smooth function, the estimate will not even be the same if computed two times with the same value of θ. A solution to this problem discussed by Andrieu et al. (2004) is to use the same noise realization each time the likelihood function is estimated. This will lead to a smooth estimate.

To use the prediction error method, it is necessary to compute the models prediction of $y(t_k)$ given Z^{k-1}, i.e., $\hat{y}(t_k|t_{k-1}, \theta)$. The predictions can for example be computed using the particle filter. Using the predictions from the particle filter (or another method), the parameters can be estimated using

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \big(y(t_k) - \hat{y}(t_k|t_{k-1}, \theta)\big)^T \Lambda^{-1} \big(y(t_k) - \hat{y}(t_k|t_{k-1}, \theta)\big) \tag{4.97a}$$

$$\hat{\theta} = \arg\min_{\theta} V_N(\theta, Z^N) \tag{4.97b}$$

where Λ weighs together the relative importance of the measurements. As for the maximum likelihood method, the result will depend on the tuning of the particle filter and the function V_N will not be smooth since the particle filter itself is stochastic. As before, the latter problem could be solved by using a fixed noise realization for all values of θ.

For the output-error case,

$$F\big(\dot{x}(t), x(t), u(t), \theta\big) = 0 \tag{4.98a}$$

$$y(t_k) = h(x(t_k), \theta) + e(t_k), \tag{4.98b}$$

the prediction error method is simplified, since the predictions $\hat{y}(t_k|t_{k-1}, \theta)$ then are obtained by simulating the system. This approach is discussed thoroughly by Schittkowski (2002).

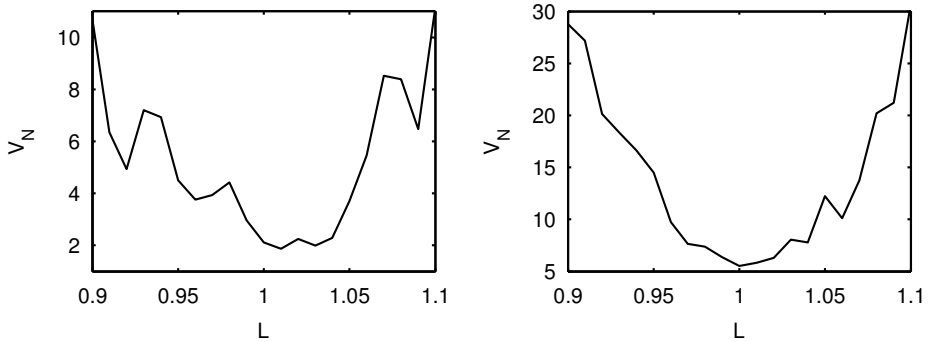
Example 4.7: Prediction error method

To illustrate the prediction error method for DAE models using particle filters, the pendulum model of Section 4.7 is considered. We have computed the criterion function V_N

in (4.97a) for the pendulum model in Section 4.7 for different values of the parameter L. The setting is the same as in Section 4.7, except for the measurement noise $e(t_k)$ in

$$y(t_k) = z_5(t_k) + e(t_k) \tag{4.99}$$

which has variance 0.01 in this example. The measurements Z^N have been obtained by simulation of the model. Measurements have been collected with sampling interval 0.05 s for the duration of 1 s. Since we have a scalar output, the weighting matrix Λ is set to 1. A straightforward implementation of the particle filter for computation of the predictions $\hat{y}(t_k|t_{k-1}, \theta)$ results in the criterion function V_N in Figure 4.4a. As can be seen in the



(a) *Different noise realizations in the particle filter for each value of L.*

(b) *The same noise realization in the particle filter for each value of L.*

Figure 4.4: *The value of the criterion function V_N for different values of the parameter L. The true value is $L = 1$.*

figure, V_N has its minimum near the true value $L = 1$. However, V_N is not very smooth which means that it would be difficult to find its minimum. To make the situation better, we compute V_N using the same noise realization in the particle filter for each value of L. The result is shown in Figure 4.4b. In this case, there is a more distinct minimum at the true value $L = 1$. Because of the approximations involved in the particle filter, V_N is not completely smooth in this case either.

If the initial condition $x(0)$ is unknown, it should normally be estimated along with the parameters. From (4.94) we get that x_1 is the state variable, and x_3 is a function of x_1. This means that only $x_1(0)$ should be parameterized and estimated along with the parameters, while $x_3(0)$ is computed from (4.94b). However, since the particle filter can work with the *distribution* of $x_1(0)$ other approaches, such as spreading the particles in an area where the $x_1(0)$ is known to lie, are possible.

There are also other parameter estimation methods available than the well-established prediction-error and maximum likelihood methods. One method which has been sug-

gested is to extend the internal variables x with the unknown parameters θ,

$$z = \begin{pmatrix} x \\ \theta \end{pmatrix} \tag{4.100}$$

and then estimate z using an estimation method like the extended Kalman filter or the particle filter (e.g., Ljung, 1979; Schön and Gustafsson, 2003; Andrieu et al., 2004). Since θ is constant, a direct implementation of the particle filter will not work (Andrieu et al., 2004, Section IV.A). Alternative methods are discussed by Andrieu et al. (2004).

4.9 Conclusions

We have presented a theoretical basis for introduction of noise processes in DAE models. The exact conditions that this gives can be hard to use in practice, for example since it requires rank tests. Therefore, an approximate solution was proposed. This solution uses the kind of DAE solvers that are included in modeling environments for object-oriented modeling. These solvers produce an approximation of the transformation that is necessary to include noise in an appropriate way.

It was also discussed how particle filtering can be implemented for DAE models. An example which shows that it is possible to implement a particle filter using a DAE model was presented. The results were similar to what could be expected from an implementation using a regular state-space model.

It was also discussed how estimation of unknown parameters can be performed using the prediction error and maximum likelihood methods. Provided that the model structure is well-defined, the particle filter can be used in the implementation.

Further research issues include to examine if it is possible to implement other estimation methods such as extended Kalman filtering for DAE models. Another research issue, which also is of interest for other model structures than DAEs, is how the particle filter should be tuned to provide smooth estimates of the probability density function $f_y(\theta, Z^N)$ and the prediction error criterion function $V_N(\theta, Z^N)$ for parameter estimation.

5

Identifiability and Observability for DAEs Based on Kunkel and Mehrmann

In this chapter we will discuss how rank tests can be used to examine local identifiability and observability for differential-algebraic equations.

5.1 Introduction

For state-space models, it is common practice to use rank tests to examine observability and identifiability. Consider for example the state-space model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.1a}$$
$$y(t) = Cx(t) \tag{5.1b}$$

with $x \in \mathbb{R}^n$. The basic way to examine observability for this system is to check if the matrix

$$\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \tag{5.2}$$

has full column rank, which is a necessary and sufficient condition (e.g., Rugh, 1996). Similarly, a sufficient condition for local identifiability and local weak observability of a nonlinear state-space system

$$\dot{x}(t) = f\big(x(t), \theta\big) \tag{5.3a}$$
$$y(t) = h\big(x(t), \theta\big) \tag{5.3b}$$

is existence of an integer k such that the matrix

$$\begin{pmatrix} h_{x,\theta}\big(x(t),\theta\big) \\ h^{(1)}_{x,\theta}\big(x(t),\theta\big) \\ \vdots \\ h^{(k)}_{x,\theta}\big(x(t),\theta\big) \end{pmatrix} \tag{5.4}$$

has full rank. As before, we use the notation

$$h^{(k)}_{x,\theta} = \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_{n_x}} & \frac{\partial}{\partial \theta_1} & \cdots & \frac{\partial}{\partial \theta_{n_\theta}} \end{pmatrix} \left(\frac{d^k}{dt^k} h\big(x(t),\theta\big) \right). \tag{5.5}$$

This test is for example discussed by Walter (1982, Section 3.2.1). In this chapter, we will discuss how such methods can be extended to DAE models. To do this, we will use the results by Kunkel and Mehrmann (2001) that were reviewed in Section 2.2. Since DAE models are more difficult to handle, the results will be more complicated than for state-space systems.

We will also show that these results can be used to derive and extend standard observability and identifiability results for state-space models.

5.2 Identifiability

Consider the nonlinear DAE

$$F\big(\dot{x}(t), x(t), \theta, u(t), t\big) = 0 \tag{5.6a}$$

$$y(t) = h\big(x(t)\big). \tag{5.6b}$$

According to the definition in Section 3.4, this DAE is locally identifiable at θ_0, x_0 for the input $u(t)$ if there exists a neighborhood V of θ_0 for which

$$\left.\begin{array}{c} \theta \in V \\ y(\theta_0, t) = y(\theta, t) \text{ for all } t \end{array}\right\} \Rightarrow \theta = \theta_0 \tag{5.7}$$

where $y(\bar{\theta}, t)$ is the output y of (5.6) with the input $u(t)$, $\theta = \bar{\theta}$, and the consistent initial condition x_0. The system is locally identifiable if it is locally identifiable at all $\theta_0 \in D_{\mathcal{M}}$.

Identifiability for nonlinear DAE models has not been treated much in literature, one reference being Ljung and Glad (1994). For state-space models, the problem is well treated. See, e.g., the book by Walter (1982).

As discussed previously, identifiability for nonlinear systems is closely related to observability. This can be seen by noting that identifiability of θ in the DAE model

$$F\big(\dot{x}(t), x(t), \theta, u(t), t\big) = 0 \tag{5.8a}$$

$$y(t) = h\big(x(t)\big) \tag{5.8b}$$

can bee seen as observability of the (constant) variable $\theta(t)$ in the model

$$F\big(\dot{x}(t), x(t), \theta(t), u(t), t\big) = 0 \tag{5.9a}$$

$$\dot{\theta}(t) = 0 \tag{5.9b}$$

$$y(t) = h\big(x(t)\big). \tag{5.9c}$$

5.3 Observability Tests Based on Kunkel and Mehrmann

We consider a nonlinear DAE,

$$G\big(\dot{x}(t), x(t), u(t), t\big) = 0 \tag{5.10a}$$

$$y(t) = h\big(x(t)\big) \tag{5.10b}$$

where $x(t) \in \mathbb{R}^{n_x}$ are the internal variables, $u(t)$ external inputs, and $y(t)$ a measured output. The idea when examining observability for the system (5.10), is that if the system is observable, then enough information should be contained in the equations to compute $x(t)$ when $u(t)$ and $y(t)$ are known signals. This means that (5.10a) and (5.10b) both should be used as equations that give information about $x(t)$. Collecting the equations gives the extended DAE

$$\underbrace{\begin{pmatrix} G\big(\dot{x}(t), x(t), u(t), t\big) \\ y(t) - h\big(x(t)\big) \end{pmatrix}}_{F} = 0. \tag{5.11}$$

What needs to be done is to examine if $x(t)$ can be solved uniquely from (5.11). Locally, the uniqueness of the solutions can be examined using the method by Kunkel and Mehrmann (2001) that is given in Theorem 2.2. Doing this results in the following theorem, which as far as we know is a novel application of the theory of Kunkel and Mehrmann.

Theorem 5.1
Assume that the extended DAE (5.11) *fulfills the conditions of Theorem 2.2 for some μ, a, d, and v and the solution $x_0(t)$. Then the original DAE* (5.10) *is locally weakly observable at $x_0(t)$ if and only if $a = n_x$ where n_x is the dimension of x.*

Note that $u(t)$ and $y(t)$ should be seen as time-dependent signals and thus be included in the general time-dependency t in Theorem 2.2.

Proof: Assume that $a = n_x$. Then, according to Theorem 2.2, the solution to the extended DAE (5.11) is locally described by

$$x_3(t) = \mathcal{R}(t) \tag{5.12}$$

and x_1 and x_2 have dimension 0. Since x_3 then describes the solution for $x(t)$, these variables are (locally) determined by the extended DAE. This means that if $y(t)$ is replaced by the output from a similar system with the solution $x_0(t)$, then there is a neighborhood of $x_0(t)$ where x is uniquely determined by (5.11) so the original DAE is locally observable.

Now assume that $a < n_x$. Then, according to Theorem 2.2, the solution to the extended DAE (5.11) is locally described by

$$\dot{x}_1(t) = \mathcal{L}\big(t, x_1(t), x_2(t), \dot{x}_2(t)\big) \tag{5.13a}$$

$$x_3(t) = \mathcal{R}\big(t, x_1(t), x_2(t)\big) \tag{5.13b}$$

where the dimension of at least one of x_1 and x_2 is greater than zero. According to Proposition 2.1 there is a neighborhood where the initial condition of these variables can be chosen freely. This means that $x(t)$ has an undetermined initial condition or is a parameter that can be varied freely without changing the output $y(t)$. This means that the original system is not locally weakly observable.

Finally, the case $a > n_x$ cannot occur since a is the difference between the ranks of the matrices in (2.46) and (2.48). □

5.4 Identifiability Tests based on Kunkel and Mehrmann

We consider a nonlinear DAE with unknown parameters,

$$G\big(\dot{z}(t), z(t), \theta, u(t), t\big) = 0 \tag{5.14a}$$

$$y(t) = h\big(z(t), \theta\big) \tag{5.14b}$$

where $z(t) \in \mathbb{R}^{n_z}$ are the internal variables, $\theta \in \mathbb{R}^{n_\theta}$ the unknown constant parameters, $u(t)$ external inputs, and $y(t)$ a measured output. As discussed in Section 5.2, identifiability of the DAE (5.14) can be seen as adding the equation

$$\dot{\theta}(t) = 0 \tag{5.15}$$

and examining if

$$x(t) = \begin{pmatrix} z(t) \\ \theta(t) \end{pmatrix} \tag{5.16}$$

is observable in the new system. We can thus use the results on observability from the previous section to examine identifiability. The extended DAE then takes the form

$$\underbrace{\begin{pmatrix} G\big(\dot{z}(t), z(t), \theta(t), u(t), t\big) \\ y(t) - h\big(z(t), \theta\big) \\ \dot{\theta}(t) \end{pmatrix}}_{F} = 0. \tag{5.17}$$

Applying the results in the previous section results in the following theorem.

Theorem 5.2
Assume that the extended DAE (5.17) *fulfills the conditions of Theorem 2.2 for some μ, a, d, and v with*

$$x(t) = \begin{pmatrix} z(t) \\ \theta(t) \end{pmatrix} \tag{5.18}$$

at $z_0(t)$, θ_0. Then the original DAE (5.14) *is locally identifiable and locally weakly observable at $z_0(t)$, θ_0 if and only if $a = n_z + n_\theta$ where n_z is the dimension of z and n_θ is the dimension of θ.*

Proof: The result follows directly from Theorem 5.1 with

$$x(t) = \begin{pmatrix} z(t) \\ \theta(t) \end{pmatrix}. \tag{5.19}$$

□

If it is known beforehand that the system is observable if all parameter values θ are known, then it is possible to examine local identifiability without having to treat observability at the same time. This is described in the following corollary.

Corollary 5.1
Assume that the original DAE (5.14) *is locally weakly observable if θ is known and that the extended DAE* (5.17) *fulfills the conditions of Theorem 2.2 for some μ, a, d, and v with*

$$x(t) = \begin{pmatrix} z(t) \\ \theta(t) \end{pmatrix} \tag{5.20}$$

at θ_0. Then the original DAE (5.14) *is locally identifiable at θ_0 if and only if $a = n_z + n_\theta$ where n_z is the dimension of z and n_θ is the dimension of θ.*

Proof: If $a = n_z + n_\theta$, the system is clearly both locally weakly observable and locally identifiable according to Theorem 5.2. If $a < n_z + n_\theta$, the solution of the extended DAE is locally described by

$$\dot{x}_1(t) = \mathcal{L}\big(t, x_1(t), x_2(t), \dot{x}_2(t)\big) \tag{5.21a}$$

$$x_3(t) = \mathcal{R}\big(t, x_1(t), x_2(t)\big) \tag{5.21b}$$

where the initial value of x_1 and x_2 can be chosen freely in some neighborhood and the dimension of at least one of these variables is greater than zero. Since $z(t)$ is locally weakly observable, it must be part of $x_3(t)$. At least part of θ is thus included in $x_1(t)$ and/or $x_2(t)$, so it is not locally identifiable. □

Example 5.1: Identifiability based on Kunkel and Mehrmann

Consider the DAE

$$\begin{pmatrix} \theta\dot{z}(t) - u(t) \\ y(t) - z(t) \end{pmatrix} = 0. \tag{5.22}$$

To examine identifiability using Theorem 5.2, we consider the extended DAE

$$\underbrace{\begin{pmatrix} \theta(t)\dot{z}(t) - u(t) \\ y(t) - z(t) \\ \dot{\theta}(t) \end{pmatrix}}_{F} = 0. \tag{5.23}$$

Let

$$x(t) = \begin{pmatrix} z(t) \\ \theta(t) \end{pmatrix}. \tag{5.24}$$

To examine if the conditions of Theorem 2.2 are satisfied, we must verify that Property 2.1 holds. We will first verify that if holds for $\mu = 2$. Not that the number of variables are $n = 2$ and the number of equations are $m = 3$. The steps of Property 2.1 can be verified as follows.

1. We have that

$$F_2 = \begin{pmatrix} F \\ \frac{d}{dt}F \\ \frac{d^2}{dt^2}F \end{pmatrix} = \begin{pmatrix} \theta(t)\dot{z}(t) - u(t) \\ y(t) - z(t) \\ \dot{\theta}(t) \\ \dot{\theta}(t)\dot{z}(t) + \theta(t)\ddot{z}(t) - \dot{u}(t) \\ \dot{y}(t) - \dot{z}(t) \\ \ddot{\theta}(t) \\ \ddot{\theta}(t)\dot{z}(t) + 2\dot{\theta}(t)\ddot{z}(t) + \theta(t)z^{(3)}(t) - \ddot{u}(t) \\ \ddot{y}(t) - \ddot{z}(t) \\ \theta^{(3)}(t) \end{pmatrix}. \tag{5.25}$$

All variables except the time t are determined by the equations, so $\mathbb{L}_2 \subseteq \mathbb{R}^9$ forms a manifold of dimension 1. This gives

$$r = 9 - 1 = 8. \tag{5.26}$$

2. We have

$$F_{2;x,\dot{x},\ddot{x},x^{(3)}} = \begin{pmatrix} 0 & \dot{z} & \theta & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddot{z} & \dot{\theta} & \dot{z} & \theta & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & z^{(3)} & \ddot{\theta} & 2\ddot{z} & 2\dot{\theta} & \dot{z} & \theta & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.27}$$

which gives that

$$\operatorname{rank} F_{2;x,\dot{x},\ddot{x},x^{(3)}} = 8 \tag{5.28}$$

provided that, for example, $\theta \neq 0$ and $\ddot{z} \neq 0$. This is consistent with $r = 8$.

3. Since $\operatorname{corank} F_{2;x,\dot{x},\ddot{x},,x^{(3)}} = 1$ and

$$F_{1;x,\dot{x},\ddot{x}} = \begin{pmatrix} 0 & \dot{z} & \theta & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \ddot{z} & \dot{\theta} & \dot{z} & \theta & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.29}$$

has full row rank ($\operatorname{corank} F_{1;x,\dot{x},\ddot{x}} = 0$), we have that $v = 1$.

4. We have

$$F_{2;\dot{x},\ddot{x},x^{(3)}} = \begin{pmatrix} \theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \dot{\theta} & \dot{z} & \theta & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \ddot{\theta} & 2\ddot{z} & 2\dot{\theta} & \dot{z} & \theta & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.30}$$

which gives

$$a = r - \operatorname{rank} F_{2;\dot{x},\ddot{x},x^{(3)}} = 8 - 6 = 2. \tag{5.31}$$

This gives that T_2 is of size zero.

5. Since T_2 is of size zero, we have that

$$\operatorname{rank} F_{\dot{x}} T_2 = 0. \tag{5.32}$$

Since

$$m - a - v = 3 - 2 - 1 = 0, \tag{5.33}$$

we have a well-defined

$$d = 0. \tag{5.34}$$

Corresponding calculations give that Property 2.1 is fulfilled also with $\mu = 3$, $a = 2$, $d = 0$, and $v = 1$. This gives that Theorem 2.2 is fulfilled for the extended DAE. Since $a = n_z + n_\theta$, Theorem 5.2 gives that the model is locally identifiable and locally weakly observable.

The calculations in the example involve quite large matrices, but the fact that the smaller matrices are submatrices of $F_{2;x,\dot{x},\ddot{x},x^{(3)}}$ makes the computations easier.

5.5 Application to State-Space Models

As was mentioned in the beginning of the chapter, the method discussed here can be seen as a generalization of rank tests that are used to examine identifiability and observability of state-space models. To make this connection clear, we present results from state-space theory in this section and prove them with the general theory of Section 5.3 and 5.4. We will also see that the methods discussed here will make it possible to show that results that usually are referred to as necessary conditions, under certain conditions are necessary and sufficient.

First consider a linear state-space model,

$$\dot{z}(t) = Az(t) \tag{5.35a}$$

$$y(t) = Cz(t) \tag{5.35b}$$

where $z \in \mathbb{R}^{n_z}$, $n_z \geq 1$, and $y \in \mathbb{R}^{n_y}$, $n_y \geq 1$. We will show that observability of this model is equivalent to the basic observability test (5.2). To examine observability, we will use Theorem 5.1. The extended DAE is

$$\underbrace{\begin{pmatrix} \dot{z}(t) - Az(t) \\ Cz(t) - y(t) \end{pmatrix}}_{F} = 0, \tag{5.36}$$

which gives the $(n_z + n_y)(\mu + 1) \times (\mu + 2)n_z$ matrix $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ (with $x(t) = z(t)$)

$$F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}} = \begin{pmatrix} -A & I & 0 & \cdots & 0 \\ C & 0 & 0 & & \vdots \\ 0 & -A & \ddots & 0 & 0 \\ \vdots & & 0 & -A & I \\ 0 & \cdots & 0 & C & 0 \end{pmatrix}. \tag{5.37}$$

Note that $m = n_z + n_y$ and $n = n_z$. By multiplying this matrix from the right with the full rank matrix

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 \\ A & I & 0 & & 0 \\ A^2 & A & I & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ A^{\mu+1} & A^{\mu} & \cdots & A & I \end{pmatrix} \tag{5.38}$$

and then from the left with the full rank matrix

$$\begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & I & 0 & 0 & 0 & & 0 \\ -C & 0 & 0 & I & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & I & 0 & & 0 \\ -CA & 0 & -C & 0 & 0 & I & & 0 \\ \vdots & & & & & & \ddots & \vdots \\ -CA^{\mu-1} & 0 & -CA^{\mu-2} & 0 & & \cdots & \cdots & I \end{pmatrix} \tag{5.39}$$

$F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ can be brought into the form

$$\begin{pmatrix} 0 & I & 0 & 0 & \cdots & 0 \\ C & 0 & 0 & 0 & & 0 \\ 0 & 0 & I & 0 & & 0 \\ CA & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & I & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & I \\ CA^{\mu} & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{5.40}$$

By row permutations, this matrix can be written as

$$\begin{pmatrix} C & 0 \\ CA & 0 \\ \vdots & 0 \\ CA^{\mu} & 0 \\ 0 & I \end{pmatrix}. \tag{5.41}$$

Since multiplication with full rank matrices and row permutations do not change the rank of a matrix, $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ has full column rank if and only if

$$\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\mu} \end{pmatrix}. \tag{5.42}$$

has full rank.

It must now be shown that Property 2.1 is fulfilled for $\mu = n_z$ and the value of a must be determined to see if the model is locally weakly observable.

Let $\mu = n_z$ and assume that $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ has full column rank so that $r = (\mu + 2)n_z$. According to the Cayley-Hamilton theorem, $F_{\mu-1;x,\dot{x},\ldots,x^{(\mu)}}$ also has full column rank, so

$$v = \big((n_z + n_y)(\mu + 1) - (\mu + 2)n_y\big) - \big((n_z + n_y)\mu - (\mu + 1)n_z\big) = n_y. \tag{5.43}$$

Furthermore, $a = n_z$ since $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ and $F_{\mu;\dot{x},\ldots,x^{(\mu+1)}}$ have full column rank. This gives that T_2 is the empty matrix, so $d = 0$. Also, $m - a - v = 0$, so Property 2.1 is satisfied with $a = n_z$ for $\mu = n_z$.

Now assume that $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ does not have full rank, so that $r = (\mu+2)n_z - \Delta r$ for some $\Delta r > 0$. According to the Cayley-Hamilton theorem we also have

$$\operatorname{rank} F_{\mu-1;x,\dot{x},\ldots,x^{(\mu)}} = (\mu + 1)n_z - \Delta r, \tag{5.44}$$

so

$$v = \big((n_z+n_y)(\mu+1)-(\mu+2)n_z+\Delta r\big)-\big((n_z+n_y)\mu-(\mu+1)n_z+\Delta r\big) = n_y. \tag{5.45}$$

Now, $a = n_z - \Delta r$ since $F_{\mu;\dot{x},\ldots,x^{(\mu+1)}}$ has full column rank by construction. Also, $d = \Delta r$ since $F_{\dot{x}}$ has full column rank by construction. This gives that $m - a - v = \Delta r$ so Property 2.1 is satisfied.

The above discussion also holds for $\mu = n_z + 1$ according to the Cayley-Hamilton theorem. This gives that the conditions of Theorem 2.2 are satisfied, so Corollary 5.1 gives that the model is locally weakly observable if and only if

$$\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n_z} \end{pmatrix} \tag{5.46}$$

has full column rank. According to the Cayley-Hamilton theorem, this is equivalent to full rank of the matrix

$$\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n_z-1} \end{pmatrix} \tag{5.47}$$

Full rank of this matrix is a standard observability criterion for linear state-space systems.

If the model has multiple outputs ($n_y \geq 2$), it may be sufficient to consider a matrix with fewer rows. This is discussed below in connection with the so-called observability indices. Also note that for linear model, local weak observability is equivalent to global observability since all equations involved are linear.

It is well known that full rank of the matrix (5.47) is a necessary and sufficient condition for observability of a linear state-space model. For nonlinear state-space models the situation is not as simple as for the linear case. While similar rank tests exist, they only give sufficient conditions for observability. The following theorem shows how the method that is discussed in this chapter not only reduces to a standard rank test for observability of nonlinear state-space models, but also shows what conditions that need to be satisfied to make it a necessary and sufficient condition.

In the theorem, we use the notation $h_z^{(k)}\big(z(t)\big)$ for the partial derivatives with respect to z of the k:th time derivative of the function $h\big(z(t)\big)$,

$$h_z^{(k)}\big(z(t)\big) = \begin{pmatrix} \frac{\partial}{\partial z_1} & \cdots & \frac{\partial}{\partial z_{n_z}} \end{pmatrix} \left(\frac{d^k}{dt^k} h\big(z(t)\big) \right). \tag{5.48}$$

Note that for the state-space model

$$\dot{z}(t) = f\big(z(t)\big) \tag{5.49a}$$

$$y(t) = h\big(z(t)\big), \tag{5.49b}$$

the time derivatives can be recursively defined by

$$h^{(0)}\big(z(t)\big) = h\big(z(t)\big) \tag{5.50a}$$

$$h^{(i+1)}\big(z(t)\big) = h_z^{(i)}\big(z(t)\big) f\big(z(t)\big). \tag{5.50b}$$

Theorem 5.3
The nonlinear state-space model

$$\dot{z}(t) = f\big(z(t)\big) \tag{5.51a}$$

$$y(t) = h\big(z(t)\big) \tag{5.51b}$$

with $z \in \mathbb{R}^{n_z}$, $n_z \geq 1$, and $y \in \mathbb{R}^{n_y}$, $n_y \geq 1$, is locally weakly observable if and only if the matrix

$$\begin{pmatrix} h_z\big(z(t)\big) \\ h_z^{(1)}\big(z(t)\big) \\ \vdots \\ h_z^{(\mu)}\big(z(t)\big) \\ h_z^{(\mu+1)}\big(z(t)\big) \end{pmatrix} \tag{5.52}$$

has full column rank if μ is chosen so that the last two block rows, $h_z^{(\mu)}\big(z(t)\big)$ and $h_z^{(\mu+1)}\big(z(t)\big)$, do not add column rank.

The condition of this theorem is typically referred to as a *sufficient* condition for observability (e.g., Nijmeijer and van der Schaft, 1990; Isidori, 1989). This theorem extends the standard sufficient condition to a *necessary and sufficient* condition.

Proof: The extended DAE is

$$\underbrace{\begin{pmatrix} \dot{z}(t) - f\big(z(t)\big) \\ h\big(z(t)\big) - y(t) \end{pmatrix}}_{F} = 0. \tag{5.53}$$

Note that the time derivatives of $f\big(z(t)\big)$ can be defined recursively by

$$f^{(0)}\big(z(t)\big) = f\big(z(t)\big) \tag{5.54a}$$

$$f^{(i+1)}\big(z(t)\big) = f_z^{(i)}\big(z(t)\big) f\big(z(t)\big). \tag{5.54b}$$

This gives the $(n_z + n_y)(\mu + 1) \times (\mu + 2)n_z$ matrix $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ (with $x(t) = z(t)$)

$$F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}} = \begin{pmatrix} f_z\big(z(t)\big) & I & 0 & \cdots & 0 & 0 \\ h_z\big(z(t)\big) & 0 & 0 & & & \vdots \\ f_z^{(1)}\big(z(t)\big) & 0 & I & & & \vdots \\ h_z^{(1)}\big(z(t)\big) & 0 & 0 & \ddots & & \vdots \\ \vdots & & & & 0 & 0 \\ f_z^{(\mu)}\big(z(t)\big) & 0 & 0 & \cdots & 0 & I \\ h_z^{(\mu)}\big(z(t)\big) & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{5.55}$$

We have that $m = n_z + n_y$ and $n = n_z$. Through column operations that do not change the rank, this matrix can be brought into the form

$$\begin{pmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ h_z\big(z(t)\big) & 0 & 0 & & & \vdots \\ 0 & 0 & I & & & \vdots \\ h_z^{(1)}\big(z(t)\big) & 0 & 0 & \ddots & & \vdots \\ \vdots & & & & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I \\ h_z^{(\mu)}\big(z(t)\big) & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{5.56}$$

This matrix has full column rank if and only if the matrix

$$\begin{pmatrix} h_z\big(z(t)\big) \\ h_z^{(1)}\big(z(t)\big) \\ \vdots \\ h_z^{(\mu-1)}\big(z(t)\big) \\ h_z^{(\mu)}\big(z(t)\big) \end{pmatrix} \tag{5.57}$$

has full column rank.

It must now be examined if Property 2.1 is fulfilled and the value of a must be determined to see if the model is locally weakly observable. Let μ be selected so that the block row $h_z^{(\mu)}\big(z(t)\big)$ in (5.57) does not add rank to (5.57). Such a μ always exists since the maximum rank of (5.57) is n_z.

First assume that $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ has full column rank so that $r = (\mu+2)n_z$. Since the block row $h_z^{(\mu)}\big(z(t)\big)$ in (5.57) does not add column rank, $F_{\mu-1;x,\dot{x},\ldots,x^{(\mu)}}$ also has full column rank, so

$$v = \big((n_z+n_y)(\mu+1) - (\mu+2)n_y\big) - \big((n_z+n_y)\mu - (\mu+1)n_z\big) = n_y. \tag{5.58}$$

Furthermore, $a = n_z$ since $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ and $F_{\mu;\dot{x},\ldots,x^{(\mu+1)}}$ have full column rank. This gives that T_2 is the empty matrix, so $d = 0$. Also, $m - a - v = 0$, so Property 2.1 is satisfied with $a = n_z$.

Now assume that $F_{\mu;x,\dot{x},\ldots,x^{(\mu+1)}}$ does not have full rank, so that $r = (\mu+2)n_z - \Delta r$ for some $\Delta r > 0$. Since the last block row in (5.57) does not add column rank, we have

$$\operatorname{rank} F_{\mu-1;x,\dot{x},\ldots,x^{(\mu)}} = (\mu+1)n_z - \Delta r, \tag{5.59}$$

so

$$v = \big((n_z+n_y)(\mu+1) - (\mu+2)n_z + \Delta r\big) - \big((n_z+n_y)\mu - (\mu+1)n_z + \Delta r\big) = n_y. \tag{5.60}$$

Now, $a = n_z - \Delta r$ since $F_{n_z;\dot{x},\ldots,x^{(n_z+1)}}$ has full column rank by construction. Also, $d = \Delta r$ since $F_{\dot{x}}$ has full column rank by construction. Since $m - a - v = \Delta r$ we have that Property 2.1 is satisfied with $a = n_z - \Delta r$.

The above discussion also holds for μ replaced by $\mu + 1$ since the last block row $h_z^{(\mu+1)}\big(z(t)\big)$ also does not add column rank. This gives that the conditions of Theorem 2.2 are satisfied, so Corollary 5.1 gives that the system is locally weakly observable if and only if (5.52) has full column rank. □

The next result shows how the methods discussed here also can be reduced to a rank test for identifiability of state-space systems. This leads to a version of the identifiability test in Section 3.2.1 in Walter (1982) (use of the implicit function theorem to examine identifiability) if the so-called exhaustive summary is taken as the derivatives of the output. Here we show that it can be taken as a necessary and sufficient condition with appropriately selected matrix dimensions.

Corollary 5.2
The nonlinear state-space system

$$\dot{z}(t) = f\big(z(t), \theta\big) \tag{5.61a}$$
$$y(t) = h\big(z(t), \theta\big) \tag{5.61b}$$

with $z \in \mathbb{R}^{n_z}$, $n_z \geq 1$, $\theta \in \mathbb{R}^{n_\theta}$, $n_\theta \geq 1$, *and* $y \in \mathbb{R}^{n_y}$, $n_y \geq 1$, *is locally identifiable and locally weakly observable if and only if the matrix*

$$\begin{pmatrix} h_{z,\theta}\big(z(t), \theta\big) \\ h_{z,\theta}^{(1)}\big(z(t), \theta\big) \\ \vdots \\ h_{z,\theta}^{(\mu)}\big(z(t), \theta\big) \\ h_{z,\theta}^{(\mu+1)}\big(z(t), \theta\big) \end{pmatrix} \tag{5.62}$$

has full column rank, if μ *is chosen so that the last two block rows,* $h_{z,\theta}^{(\mu)}\big(z(t), \theta\big)$ *and* $h_{z,\theta}^{(\mu+1)}\big(z(t), \theta\big)$, *do not add column rank.*

Proof: The problem can be seen as examining observability of the system

$$\dot{z}(t) = f\big(z(t), \theta(t)\big) \tag{5.63}$$
$$\dot{\theta}(t) = 0 \tag{5.64}$$
$$y(t) = h\big(z(t), \theta\big). \tag{5.65}$$

Applying Theorem 5.3, we directly get that this system is locally weakly observable if and only if the matrix (5.62) has full column rank. □

5.6 Other Insights Using Kunkel's and Mehrmann's Theory

In this section we will discuss how *observability indices* and *zero dynamics* can be discussed using the theory for DAE models by Kunkel and Mehrmann (2001).

5.6.1 Observability Indices

As discussed above, a sufficient (and necessary under certain conditions) condition for local weak observability of the state-space system

$$\dot{x}(t) = f\big(x(t)\big) \tag{5.66a}$$
$$y(t) = h\big(x(t)\big) \tag{5.66b}$$

is full rank of the matrix

$$\begin{pmatrix} h_x\big(x(t)\big) \\ h_x^{(1)}\big(x(t)\big) \\ \vdots \\ h_x^{(n_x-1)}\big(x(t)\big) \\ h_x^{(n_x)}\big(x(t)\big) \end{pmatrix}. \tag{5.67}$$

where n_x is the dimension of x. This means that all outputs y are differentiated n_x times to prove identifiability. However, if y and h are vector valued, it may be sufficient to examine a smaller matrix to conclude that the system is observable. In other words, it may not be necessary to differentiate all outputs n_x times. To see this, assume that there are n_y outputs and let the i:th time derivative of output j be denoted

$$h_j^{(i)}\big(x(t)\big). \tag{5.68}$$

The partial derivatives of this time derivative with respect to the states x is then denoted

$$h_{j;x}^{(i)}\big(x(t)\big). \tag{5.69}$$

The idea is now that if it is possible to find integers $\sigma_1, \ldots, \sigma_{n_y}$ ($\sigma_k \leq n_x$) such that the matrix

$$\begin{pmatrix} h_{1;x}\big(x(t)\big) \\ h_{1;x}^{(1)}\big(x(t)\big) \\ \vdots \\ h_{1;x}^{(\sigma_1)}\big(x(t)\big) \\ h_{2;x}\big(x(t)\big) \\ \vdots \\ h_{n_y-1;x}^{(\sigma_{n_y-1})}\big(x(t)\big) \\ h_{n_y;x}\big(x(t)\big) \\ h_{n_y;x}^{(1)}\big(x(t)\big) \\ \vdots \\ h_{n_y;x}^{(\sigma_{n_y})}\big(x(t)\big) \end{pmatrix} \tag{5.70}$$

has $\operatorname{rank} = n_x$, then this is a sufficient condition for observability. That this is a sufficient condition can be realized since $\operatorname{rank} = n_x$ of (5.70) implies full rank of (5.67). This means that the first output is differentiated σ_1 times, the second output is differentiated σ_2 times, and so on.

Often there are several ways to choose the set of integers σ_k, $k = 1, \ldots, n_y$. Typically one wants to differentiate the outputs as few times as possible, so it is desirable to have as low σ_k as possible. One way to choose the σ_k is therefore to first choose the largest σ_k as small as possible, then make the next largest as small as possible and so on. If the σ_k are chosen in this way, they are called the *observability indices* of the system (e.g., Nijmeijer and van der Schaft, 1990).

The advantage of this method is that the number of differentiations of each output is minimized. The method discussed e.g., in Theorem 5.3, differentiates all outputs the same

number of times. However, if μ is chosen as small as possible in Theorem 5.3, then μ is the smallest integer such that

$$\begin{pmatrix} h_x\big(x(t)\big) \\ h_x^{(1)}\big(x(t)\big) \\ \vdots \\ h_x^{(\mu-2)}\big(x(t)\big) \\ h_x^{(\mu-1)}\big(x(t)\big) \end{pmatrix} \tag{5.71}$$

has full column rank and

$$\begin{pmatrix} h_x\big(x(t)\big) \\ h_x^{(1)}\big(x(t)\big) \\ \vdots \\ h_x^{(\mu-2)}\big(x(t)\big) \end{pmatrix} \tag{5.72}$$

does not have full column rank. This means that $\mu - 1$ is equal to the largest observability index of the system (5.66). We formulate this as a proposition.

Proposition 5.1
Assume that the state-space system

$$\dot{x}(t) = f\big(x(t)\big) \tag{5.73a}$$
$$y(t) = h\big(x(t)\big) \tag{5.73b}$$

is locally weakly observable and fulfills the conditions of Theorem 5.3. Let μ in Theorem 5.3 be taken as small as possible. Then $\mu - 1$ is equal to the largest observability index of the system.

5.6.2 Zero Dynamics

In this section we will show how the ideas presented in this chapter also can be used for examining zero dynamics. If a system is controlled so that it follows a prescribed trajectory, then the zero dynamics is the dynamics that is not prescribed by this control law. Assume for example that a DAE model

$$G\big(\dot{z}(t), z(t), u(t), t\big) = 0 \tag{5.74a}$$
$$y(t) = h\big(z(t)\big) \tag{5.74b}$$

is to follow a prescribed trajectory $y(t) = r(t)$, $t \geq 0$ and that this trajectory can be achieved by selecting an appropriate control signal $u(t)$ and possibly initial condition $z(0)$. If all elements of $z(t)$ are uniquely determined by this control law, then there are no zero dynamics. But if some elements of $z(t)$ can be given arbitrary initial conditions and are not determined by the prescribed output, then these variables form the zero dynamics of the system. The existence of zero dynamics can be examined by the methods that previously were used to examine identifiability and observability. This can be done by seeing (5.74) as a DAE where both $z(t)$ and $u(t)$ are unknown, but $y(t)$ is known, $y(t) = r(t)$. If the system is observable, then there are no zero dynamics, since both $z(t)$ and

$u(t)$ then are determined by (5.74). If the system is not observable, then there is either zero dynamics or some components of $u(t)$ are not necessary to control $y(t)$, or both. If observability is examined using Theorem 5.1, this leads to the following proposition.

Proposition 5.2
Assume that the extended DAE

$$\underbrace{\begin{pmatrix} G\big(\dot{z}(t), z(t), u(t), t\big) \\ y(t) - h\big(z(t)\big) \end{pmatrix}}_{F} = 0 \tag{5.75}$$

fulfills the conditions of Theorem 2.2 with

$$x(t) = \begin{pmatrix} z(t) \\ u(t) \end{pmatrix} \tag{5.76}$$

for some μ, a, d, and v and the solution $x_0(t)$ and that all components of $u(t)$ are uniquely determined by any control law that achieves $y(t) = r(t)$ for some function $r(t)$. Then the original DAE (5.74) *has no zero dynamics if and only if $a = n_x$ where n_x is the dimension of x.*

It can also be noted that if the extended DAE (5.75) is separated according to the discussion in Section 2.2,

$$\hat{F}_1(t, x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3) = 0 \tag{5.77a}$$

$$\hat{F}_2(t, x_1, x_2, x_3) = 0 \tag{5.77b}$$

$$x(t) = Q \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix}, \; Q \text{ permutation matrix,} \tag{5.77c}$$

then $\hat{F}_1$ describes the zero dynamics of the system. This can be seen since these equations can be locally solved to give

$$\dot{x}_1 = \mathcal{L}(t, x_1, x_2, \dot{x}_2) \tag{5.78a}$$

$$x_3 = \mathcal{R}(t, x_1, x_2) \tag{5.78b}$$

for some functions $\mathcal{L}$ and $\mathcal{R}$. The zero dynamics are thus described by (5.78a).

Example 5.2: Zero dynamics and state-space models

To compare with results in the literature, consider the state-space model

$$\dot{z}(t) = f\big(z(t)\big) + g\big(z(t)\big)u(t) \tag{5.79a}$$

$$y(t) = h\big(z(t)\big) \tag{5.79b}$$

where $z \in \mathbb{R}^{n_z}$, $u \in \mathbb{R}$, and $y \in \mathbb{R}$. For simplicity we thus let the system be single-input and single-output. Also let the $z(t)$ in the system (5.79) be observable with $u(t)$ and $y(t)$ known, so that

$$\begin{pmatrix} h_z\big(z(t)\big) \\ h_z^{(1)}\big(z(t), u(t)\big) \\ \vdots \\ h_z^{(j)}\big(z(t), u(t), \dot{u}(t), \ldots, u^{(j-1)(t)}\big) \end{pmatrix} \tag{5.80}$$

has full (constant) rank for $j = 0, \ldots, n_z$. As before, $h_z^{(j)}$ is the partial derivative with respect to z of the j:th time derivative of output. Similarly, let $h_u^{(j)}$ be the derivative with respect to u of the j:th time derivative of the output. Let $h_u^{(\mu)}$ be the first non-zero derivative so that

$$h_u^{(j)} = 0 \quad j = 0, \ldots \mu - 1 \tag{5.81a}$$

$$h_u^{(\mu)} \neq 0. \tag{5.81b}$$

We can now use Corollary 5.1 to examine observability of $z(t)$ and $u(t)$ in the model (5.79) with $y(t)$ known, and thereby also examine if there are any zero dynamics. Form the $(n_z + 1)(\mu + 1) \times (n_z + 1)(\mu + 2)$ matrix

$$F_{\mu;x,\dot{x},\ldots,x^{\mu+1}} = \begin{pmatrix} * & * & I & 0 & \cdots & 0 & 0 & 0 & 0 \\ h_z & h_u & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & * & \cdots & I & 0 & 0 & 0 \\ h_z^{(\mu-1)} & h_u^{(\mu-1)} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ * & * & * & * & \cdots & * & * & I & 0 \\ h_z^{(\mu)} & h_u^{(\mu)} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.82}$$

where $*$ represent elements whose exact form are not important and

$$x = \begin{pmatrix} z \\ u \end{pmatrix}. \tag{5.83}$$

Note that $m = n_z + 1$. By using that $h_u^{(j)} = 0$, $j = 0, \ldots, \mu - 1$ and by column operations that do not change the rank of the matrix this can be written

$$\begin{pmatrix} 0 & 0 & I & 0 & \cdots & 0 & 0 & 0 & 0 \\ h_z & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 \\ h_z^{(\mu-1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I & 0 \\ 0 & h_u^{(\mu)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.84}$$

By examining this matrix we get that

$$r = \operatorname{rank} F_{\mu;x,\dot{x},\ldots,x^{\mu+1}} = (n_z + 1)(\mu + 1) \tag{5.85}$$

and

$$\nu = 0 \tag{5.86}$$

since the matrix has full row rank. We also get

$$a = r - \operatorname{rank} F_{\mu;\dot{x},\ldots,x^{\mu+1}} = r - n_z(\mu + 1) = \mu + 1. \tag{5.87}$$

Taking

$$Z_2^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & \\ 0 & 0 & 0 & 0 & 0 & & 1 \end{pmatrix} \tag{5.88}$$

to pick out those rows in $F_{\mu;x,\dot{x},\ldots,x^{\mu+1}}$ with $h_z^{(j)}$, we get

$$Z_2^T F_{\mu;x} = \begin{pmatrix} h_z & 0 \\ h_z^{(1)} & 0 \\ \vdots & \vdots \\ h_z^{(\mu-1)} & 0 \\ h_z^{(\mu)} & h_u^{(\mu)} \end{pmatrix} \tag{5.89}$$

so the $(n_z + 1) \times (n_z - \mu)$ matrix T_2 is taken so that the first n_z rows are the orthogonal complement to the μ first rows of (5.89), which gives that

$$d = \operatorname{rank} F_{\dot{x}} T_2 = n_z - \mu \tag{5.90}$$

since $\operatorname{rank} F_{\dot{x}}$ will pick out the first n_z rows in T_2. We also have

$$m - a - v = n_z + 1 - (\mu + 1) - 0 = n_z - \mu \tag{5.91}$$

so Property 2.1 is satisfied. Similar calculations can also be performed with μ replaced by $\mu + 1$ to show that the conditions of Theorem 2.2 are fulfilled.

To conclude, we have that $\mu + 1$ variables of z and u are determined by (5.79) when y has a prescribed trajectory. Since u is used to control y, the variables that are *not* determined by the equations must be taken from z. There are $n_z - \mu$ such variables, and these variables form the zero dynamics of the controlled system since they are not determined by the control law. We see that there are no zero dynamics in the case when $\mu = n_z$, which means that the output y must be differentiated exactly n_z times for u to appear. This is in line with standard results, see e.g., Isidori (1989).

5.7 Conclusions

In this chapter, new criteria for local identifiability and local weak observability of nonlinear differential-algebraic equations have been derived using results by Kunkel and Mehrmann (2001). The inherent complexity of nonlinear differential-algebraic equations make the criteria somewhat involved, but on the other hand the generality of DAE models allows many models to fit into the framework. We have also shown that the criteria are closely related to standard identifiability and observability criteria for state-space models, and even extend these results in some cases.

We have also discussed how zero dynamics can be examined using the methods by Kunkel and Mehrmann (2001).

6

Identifiability Tests Using Differential Algebra for Component-Based Models

This chapter discusses how the structure in component-based models can be used when examining identifiability. We will show the interesting fact that once identifiability has been examined for the components of a model, identifiability of the complete model can be examined using a reduced number of equations.

6.1 Introduction

In Section 3.4, a general method by Ljung and Glad (1994) for examining identifiability in linear and nonlinear systems, both state-space systems and differential-algebraic equations, was summarized. This method uses differential algebra which suffers from high computational complexity, and can therefore only handle quite small systems. This chapter discusses how the modularized structure in component-based models can be used to speed up the computations. Since modeling tools such as Modelica are based on component-based modeling, the approach can be useful for models created using such tools.

As discussed in Section 2.1.1, a component-based model consists of a number of components, with equations describing them, and a number of equations describing the connections between the components. Since the components represent different physical parts of the system, it is natural that they have independent parameters so that will be assumed in the present chapter. As before (Section 2.1.1), the equations describing a model with m components are written as

$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \qquad i = 1, \ldots, m. \tag{6.1}$$

Here, $l_i(t) \in \mathbb{R}^{n_{l_i}}$ are internal variables, $w_i(t) \in \mathbb{R}^{n_{w_i}}$ external variables that are used in the connections and $\theta_i \in \mathbb{R}^{\theta_i}$ unknown parameters, all in component i. As before p is the

differentiation operator with respect to time,

$$px(t) = \frac{dx(t)}{dt}. \tag{6.2}$$

With $f_i(\cdot) \in \mathbb{R}^{n_{f_i}}$, it is assumed that $n_{f_i} \geq n_{l_i}$ so that there are at least as many equations as internal variables for each component. The equations describing the connections are written

$$g\big(u(t), w(t)\big) = 0, \quad w(t) = \begin{pmatrix} w_1(t) \\ \vdots \\ w_m(t) \end{pmatrix} \tag{6.3}$$

where $u(t)$ is an external input signal. Measured output signals are specified as

$$y(t) = h\big(w(t)\big) \tag{6.4}$$

where we have assumed that no unknown parameters are included in the measurement equation. Parameters in the measurement equation could instead be handled by introducing extra components that for example scale the measured output. To summarize, a complete component-based model consists of the equations for the components, for the connections, and for the measurements,

$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \qquad i = 1, \ldots, m. \tag{6.5a}$$

$$g\big(u(t), w(t)\big) = 0 \tag{6.5b}$$

$$y(t) = h\big(w(t)\big). \tag{6.5c}$$

Identifiability of this model can be analyzed using the method described in Section 3.4. However, our main idea is to separate the identifiability analysis into two stages. The first stage is to rewrite the model for a single component using the technique given by (3.21), and thus avoiding this computation for the complete model. The second stage is to examine identifiability by combining the transformed equations for each component. For the first stage, we thus assume that the model

$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \tag{6.6}$$

can be rewritten in the equivalent form

$$\begin{aligned} A_{i,1}(w_i, p) &= 0 \\ &\vdots \\ A_{i,n_{w_i}}(w_i, p) &= 0 \\ B_{i,1}(w_i, \theta_{i,1}, p) &= 0 \\ B_{i,2}(w, \theta_{i,1}, \theta_{i,2}, p) &= 0 \\ &\vdots \\ B_{i,n_{\theta_i}}(w_i, \theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,n_{\theta_i}}, p) &= 0 \\ C_{i,1}(w_i, \theta_i, l_i, p) &= 0 \\ &\vdots \\ C_{i,n_{l_i}}(w_i, \theta_i, l_i, p) &= 0. \end{aligned} \tag{6.7}$$

Note that if the original DAE only has polynomial equations, this transformation is always possible. The A_i equations are relations that the external variables must satisfy, regardless of the value of the parameters θ. The B_i equations can be used to determine identifiability of the parameters if the w_i are known, and forms a linear regression for the parameters if the component is globally identifiable and the equations are polynomial. The C_i equations give relations for the internal variable l_i and are of no further interest in this chapter.

An important part of the model for the analysis below is the set of $A_{i,j}$. These relations between the connecting variables are independent of the choice of the parameters.

In the examples below, we discuss how the form (6.7) can be calculated for a model of a capacitor, a model of an inductor and a model of a nonlinear resistor.

Example 6.1: Capacitor component

Consider a capacitor described by the voltage drop w_1, current w_2 and capacitance θ_1. It is then described by (6.6) with

$$f_1 = \begin{pmatrix} \theta_1 \dot{w}_1 - w_2 \\ \dot{\theta}_1 \end{pmatrix}. \tag{6.8}$$

If we consider only situations where $\dot{w}_1 \neq 0$ we get the following series of equivalences.

$$\theta_1 \dot{w}_1 - w_2 = 0, \quad \dot{\theta}_1 = 0, \quad \dot{w}_1 \neq 0$$
$$\Leftrightarrow$$
$$\theta_1 \dot{w}_1 - w_2 = 0, \quad \theta_1 \ddot{w}_1 - \dot{w}_2 = 0, \quad \dot{w}_1 \neq 0$$
$$\Leftrightarrow$$
$$\theta_1 \dot{w}_1 - w_2 = 0, \quad \theta_1 \dot{w}_1 \ddot{w}_1 - \dot{w}_1 \dot{w}_2 = 0, \quad \dot{w}_1 \neq 0$$
$$\Leftrightarrow$$
$$\theta_1 \dot{w}_1 - w_2 = 0, \quad w_2 \ddot{w}_1 - \dot{w}_1 \dot{w}_2 = 0, \quad \dot{w}_1 \neq 0$$

With the notation (6.7) we thus have

$$A_{1,1} = w_2 \ddot{w}_1 - \dot{w}_1 \dot{w}_2 \tag{6.9a}$$
$$B_{1,1} = \theta_1 \dot{w}_1 - w_2 \tag{6.9b}$$

and the function s_1 of (3.22) is $\dot{w}_1$.

Example 6.2: Inductor component

Next consider an inductor where w_1 the voltage, w_2 is the current, and θ_1 the inductance. It is described by

$$f_2 = \begin{pmatrix} \theta_1 \dot{w}_2 - w_1 \\ \dot{\theta}_1 \end{pmatrix}. \tag{6.10}$$

Calculations similar to those of the previous example show that this is equivalent to

$$A_{2,1} = \ddot{w}_2 w_1 - \dot{w}_2 \dot{w}_1 \tag{6.11a}$$
$$B_{2,1} = \theta_1 \dot{w}_2 - w_1 \tag{6.11b}$$

provided $\dot{w}_2 \neq 0$.

As discussed earlier, the transformation to (6.7) can always be performed for polynomial DAEs. To show that calculations of this type in some cases also can be done for non-polynomial models, we consider a nonlinear resistor where the voltage drop is given by an arbitrary function.

Example 6.3: Nonlinear resistor component

Consider a nonlinear resistor with the equation

$$w_1 = R(w_2, \theta_1) \tag{6.12}$$

where it is assumed that the parameter θ_1 can be uniquely solved from (6.12) if the voltage w_1 and the current w_2 are known, so that

$$\theta_1 = \phi(w_1, w_2) \tag{6.13}$$

for some function ϕ. Differentiating (6.12) once with respect to time and inserting (6.13) gives

$$\dot{w}_1 = R_{w_2}\big(w_2, \phi(w_1, w_2)\big)\dot{w}_2 \tag{6.14}$$

which is a relation between the external variables w_1 and w_2. We use the notation R_x for the partial derivative of R with respect to the variables x. We thus get

$$A_{3,1} = R_{w_2}\big(w_2, \phi(w_1, w_2)\big)\dot{w}_2 - \dot{w}_1 \tag{6.15a}$$
$$B_{3,1} = \theta_1 - \phi(w_1, w_2). \tag{6.15b}$$

In the special case with a linear resistor, where $R = \theta_1 \cdot w_2$, $A_{3,1}$ reduces to

$$\dot{w}_1 = \frac{w_1}{w_2}\dot{w}_2 \tag{6.16a}$$
$$\Leftrightarrow w_2\dot{w}_1 = w_1\dot{w}_2 \tag{6.16b}$$

(assuming $w_2 \neq 0$).

6.2 Main Results

The main results of this chapter concern how the modularized structure of component-based models can be used to examine identifiability in an efficient way.

Assume that all components are identifiable if the external variables w_i of each component are measured. This means, that given measurements of

$$w_i \qquad i = 1, \ldots, m \tag{6.17}$$

the unknown parameters θ can be computed uniquely from the B polynomials. When examining identifiability of the connected system it is not a big restriction to assume that the individual components are identifiable since information is removed when not all w_i are measured. (Recall that all components have unique parameters.)

When the components have been connected, the only knowledge available about the w_i is the A polynomials and the equations $g\big(u(t), w(t)\big) = 0$ and $y(t) = h\big(w(t)\big)$. The connected system is thus identifiable if the w_i can be computed from

$$A_{ij}\big(w_i(t), p\big) = 0 \quad \begin{cases} i = 1, \ldots, m \\ j = 1, \ldots, n_{A_i} \end{cases} \tag{6.18a}$$

$$g\big(u(t), w(t)\big) = 0 \tag{6.18b}$$

$$y(t) = h\big(w(t)\big) \tag{6.18c}$$

when $u(t)$ and $y(t)$ are known. Note that this means that all $w(t)$ are algebraic variables (not differential), so that no initial conditions can be specified for any component of $w(t)$. If, on the other hand, there are several solutions to the equations (6.18) then these different solutions can be inserted into the B polynomials, so there are also several possible parameter values. In this case the connected system is therefore not identifiable.

The result is formalized in the following theorems. Note that the distinction between global and local identifiability was not discussed above, but this will be done below.

6.2.1 Global Identifiability

Global identifiability means that there is a unique solution to the identification problem, given that the measurements are informative enough. For a component (6.6) that can be rewritten in the form (6.7) global identifiability means that the $B_{i,j}$ can be solved uniquely to give the $\theta_{i,j}$. In other words there exist functions ψ that can in principle be calculated from the $B_{i,j}$, such that

$$\theta_i = \psi_i(w_i, p). \tag{6.19}$$

When the DAE consists of polynomial equations, the ψ are formed from linear regressions,

$$P_i(w_i, p)\theta_i - Q_i(w_i, p) = 0. \tag{6.20}$$

We have the following formal result on identifiability.

Theorem 6.1
Consider a component-based model where the components (6.6) *are globally identifiable with w_i measured and thus can be described in the form* (6.19). *A sufficient condition for the total model to be globally identifiable is that* (6.18) *is observable with respect to the w_i. If all the functions ψ_i of* (6.19) *are injective then this condition is also necessary.*

Proof: If (6.18) gives a global solution for $w(t)$, then this solution can be inserted into the B polynomials to give a global solution for θ since the components are globally identifiable. The connected system is thus globally identifiable. If there are several solutions for w_i and the functions ψ_i of (6.19) are injective, then there are also several solutions for θ, so the system is not globally identifiable since the identification problem has more than one solution. □

6.2.2 Local Identifiability

Local identifiability of a model structure means that locally there is a unique solutions to the identification problem, but globally there may be more than one solution. This means that the description (6.19) is valid only locally. We get the following result on local identifiability.

Theorem 6.2
Consider a component-based model where the components (6.6) *are locally identifiable with* w_i *measured and thus can be locally described in the form* (6.19). *A sufficient condition for the total model to be locally identifiable is that* (6.18) *is observable with respect to the* w_i. *If all the functions* ψ_i *of* (6.19) *are locally injective then this condition is also necessary.*

Proof: If (6.18) gives a locally unique solution for $w(t)$, then this solution can be inserted into the B polynomials to give a local solution for θ since the components are locally identifiable. The connected system is thus locally identifiable. If there locally are several solutions for w_i and the functions ψ_i of (6.19) are injective, then there are also several local solutions for θ, so the system is not locally identifiable since the identification problem locally has more than one solution. □

6.3 Applying the Results

The techniques discussed above are intended to be used when examining identifiability for component-based models. Since each component must be transformed into the form (6.7), the first step is to perform these transformations using, e.g., differential algebra (Ljung and Glad, 1994). The transformed version of the components can then be stored together with the original model equations in model libraries. As the transformation is calculated once and for all, it should also be possible to use other methods than differential algebra to make the transformation into the form (6.7). As mentioned above, this could make it possible to handle systems described by non-polynomial differential-algebraic equations.

When a component-based model has been composed of components for which the transformation into the form (6.7) is known, identifiability of the complete model,

$$f_i\big(l_i(t), w_i(t), \theta_i, p\big) = 0 \qquad i = 1, \ldots, m. \tag{6.21a}$$

$$g\big(u(t), w(t)\big) = 0 \tag{6.21b}$$

$$y(t) = h\big(w(t)\big) \tag{6.21c}$$

can be checked by examining the solutions of the differential-algebraic equation (6.18),

$$A_{ij}\big(w_i(t), p\big) = 0 \qquad \begin{cases} i = 1, \ldots, m \\ j = 1, \ldots, n_{w_i} \end{cases} \tag{6.22a}$$

$$g\big(u(t), w(t)\big) = 0 \tag{6.22b}$$

$$y(t) = h\big(w(t)\big). \tag{6.22c}$$

The number of solutions to this differential-algebraic equation then determines if the system is identifiable, as discussed in Theorem 6.1 and 6.2. Note that the number of solutions

could vary with t, so that the system is identifiable at only some time instances. The number of solutions of the differential-algebraic equation (6.22) could be checked in different ways, and some are listed below.

Differential Algebra

If the system equations are polynomial, then one way to check the number of solutions is to use differential algebra in a similar way as was done to achieve the form (6.7). This method can be slow in some cases, but it always gives definite answers. However, in some cases this approach should be faster than to derive the transformation to the form (6.7) for the complete component-based model. Differential algebra can be used to examine both local and global identifiability, but requires that the equations are polynomial.

Kunkel & Mehrmann's Test

The analysis method by Kunkel and Mehrmann (2001) that is discussed in Section 2.2 examines the properties of nonlinear differential-algebraic equations through certain rank tests. Among other things, it is possible to use these results for examining local observability, as discussed in Section 5.3. One possibility to examine observability of (6.22) is thus to use the results in Section 5.3.

Manual Inspection

For smaller models it may be possible to examine the solvability of (6.22) by inspection of the equations and manual calculations. This can of course not be developed into a general procedure, but may still be a good approach in some cases. Manual inspection can be used to check both local and global identifiability.

6.4 Examples

In this section the techniques described in the chapter are exemplified on a very small model library consisting of a resistor model, an inductor model, and a capacitor model. Note that these components have corresponding components for example within mechanics and fluid systems. (Compare bond graphs, where generic components are used to model phenomena from all these fields.) In this small example, all variables are external.

The transformation into the form (6.7) was performed in Examples 6.1, 6.2, and 6.3, so we shall here examine the identifiability of different connections of the components. In the first example we consider the connection of a resistor and an inductor in series.

Example 6.4: Resistor and inductor in series

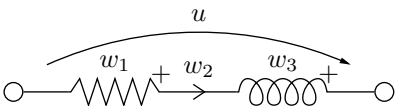


Figure 6.1: *A resistor and an inductor connected in series.*

Consider a nonlinear resistor and an inductor connected in series where the current $w_2 = f$ and total voltage u are measured as shown in Figure 6.1. Denote the voltage over the resistor with w_1 and the voltage over the inductor with w_3. Using Examples 6.2 and 6.3 we get the equations

$$\dot{w}_1 = R_{w_2}\big(w_2, \phi(w_1, w_2)\big)\dot{w}_2 \tag{6.23a}$$

$$\ddot{w}_2 w_3 = \dot{w}_2 \dot{w}_3 \tag{6.23b}$$

for the components. The connection is described by the equation

$$w_1 + w_3 = u. \tag{6.23c}$$

Differentiating the last equation once gives

$$\dot{w}_1 + \dot{w}_3 = \dot{u}. \tag{6.23d}$$

The system of equations (6.23) (with w_1, $\dot{w}_1$, w_3, and $\dot{w}_3$ as unknowns) has the Jacobian

$$\begin{pmatrix} -R_{w_2,w_1}\dot{w}_2 & 1 & 0 & 0 \\ 0 & 0 & \ddot{w}_2 & -\dot{w}_2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{6.24}$$

where

$$R_{w_2,w_1} = \frac{\partial}{\partial w_1}\Big(R_{w_2}\big(w_2, \phi(w_1, w_2)\big)\Big). \tag{6.25}$$

The Jacobian has the determinant $-R_{w_2,w_1} \cdot \dot{w}_2^2 + \ddot{w}_2$, so the system of equations is solvable for most values of the external variables. This means that the system is locally identifiable.

In the next example, two capacitors are connected in series.

Example 6.5: Two capacitors in series

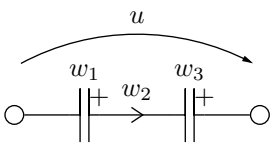


Figure 6.2: *Two capacitors connected in series.*

Now consider two capacitors connected in series where the current $w_2 = f$ and total voltage u are measured as shown in Figure 6.2. Denote the voltages over the capacitors with w_1 and w_3 respectively. Using Example 6.1 we get the equations

$$w_2\ddot{w}_1 = \dot{w}_1\dot{w}_2 \tag{6.26a}$$

$$w_2\ddot{w}_3 = \dot{w}_3\dot{w}_2 \tag{6.26b}$$

for the components and the equation

$$w_1 + w_3 = u \tag{6.27}$$

for the connections. These equations directly give that if

$$w_1(t) = \phi_1(t) \tag{6.28a}$$
$$w_3(t) = \phi_3(t) \tag{6.28b}$$

is a solution, then so are all functions of the form

$$w_1(t) = (1 + \lambda)\phi_1(t) \tag{6.29a}$$
$$w_3(t) = \phi_3(t) - \lambda\phi_1(t) \tag{6.29b}$$

for scalar λ. Since (6.9b) implies that the capacitance is an injective function of the derivative of the voltage, this shows that the system is not identifiable.

6.5 A Mechanics Model Library

In this section we will further exemplify the methods discussed in the chapter by making the transformation into the form (6.7) for all continuous components in the Modelica model library *Modelica.Mechanics.Translational* (Fritzson, 2004). This library contains the following components for one-dimensional movement, such as masses and springs.

SlidingMass models a mass moving in one dimension.

Stop models a mass that hits a stop such as a wall.

Rod models a massless rod.

Spring models a one-dimensional spring.

Damper models damping.

SpringDamper models a spring and a damper connected in parallel.

ElastoGap models a spring damper in combination with a gap.

Position models a control input to position.

Accelerate models a control input to acceleration.

Fixed models a point that is fixed.

Force models a controlled force input.

RelativeStates is used for different coordinate systems in different parts of a model.

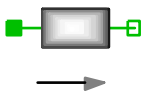
Examples contain a set of example models that use the components.

Interfaces are base models that are used when defining the components.

Sensors are used to model measurements.

The components Position, Accelerate, Fixed, Force, and Sensors are used to model inputs and outputs, and are therefore included in the connection equations (6.3). RelativeStates is also assumed to be included among the connections. Therefore the equations describing these equations do not need to be transformed. The components Stop and ElastoGap contain discontinuous dynamics and cannot be handled by the theory presented here. The components we will consider are therefore SlidingMass, Rod, Spring, Damper, and SpringDamper. First consider the SlidingMass component.

Example 6.6: SlidingMass



The SlidingMass is a component that describes a mass that slides along a surface without friction. It is described by the equation

$$m\ddot{s}(t) = f_1(t) + f_2(t) \tag{6.30}$$

where the position s and forces f_1 and f_2 are external variables and the mass m is a parameter. There are no internal variables. Applying Ritt's algorithm to this equation gives

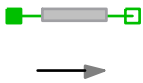
$$A = (f_1 + f_2)s^{(3)} - (\dot{f}_1 + \dot{f}_2)s^{(2)} \tag{6.31a}$$

$$B = m\ddot{s} - (f_1 + f_2) \tag{6.31b}$$

which is in the desired form (6.7). The component is globally identifiable.

Next, consider the Rod component.

Example 6.7: Rod



The Rod component describes a rod without mass. It translates the force one end to its other end. It is described by the equation

$$f_1(t) + f_2(t) = 0 \tag{6.32}$$

where the forces f_1 and f_2 are external variables. There are no internal variables or parameters, so this is already in the form (6.7) with

$$A = f_1 + f_2. \tag{6.33}$$

Now consider the Spring component.

Example 6.8: Spring

The spring component models an ideal linear spring. The equation describing it is

$$f(t) = c\big(s_1(t) - s_2(t)\big) \tag{6.34}$$

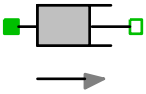
where the force f and positions of each end of the spring s_1 and s_2 are external variables. The spring constant c is a parameter. There are no internal variables. Applying Ritt's algorithm gives

$$A = (\dot{s}_1 - \dot{s}_2)f - (s_1 - s_2)\dot{f} \tag{6.35a}$$
$$B = f - c(s_1 - s_2) \tag{6.35b}$$

which is in the form (6.7). The component is globally identifiable.

Next consider the Damper.

Example 6.9: Damper



The Damper component models a linear damper. It is described by the equation

$$f(t) = d\big(\dot{s}_1(t) - \dot{s}_2(t)\big). \tag{6.36}$$

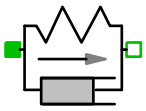
The force f and position s are external variables and the spring constant d is a parameter. There are no internal variables. Ritt's algorithm gives the form (6.7),

$$A = (\ddot{s}_1 - \ddot{s}_2)f - (\dot{s}_1 - \dot{s}_2)\dot{f} \tag{6.37a}$$
$$B = f - d(\dot{s}_1 - \dot{s}_2). \tag{6.37b}$$

The component is globally identifiable.

Finally, we consider the SpringDamper component.

Example 6.10: SpringDamper



This component which represents a spring and a damper connected in parallel, is described by the equation

$$f(t) = c\big(s_1(t) - s_2(t)\big) + d\big(\dot{s}_1(t) - \dot{s}_2(t)\big). \tag{6.38}$$

Ritt's algorithm gives

$$A = \ddot{f}(\dot{s}_1 - \dot{s}_2)^2 - (\dot{s}_1 - \dot{s}_2)\dot{f}(\ddot{s}_1 - \ddot{s}_2) - (\dot{s}_1 - \dot{s}_2)f(s_1^{(3)} - s_2^{(3)}) + \dot{f}(s_1^{(3)} - s_2^{(3)})(s_1 - s_2) + f(\ddot{s}_1 - \ddot{s}_2)^2 - (\ddot{s}_1 - \ddot{s}_2)(s_1 - s_2)\ddot{f} \tag{6.39a}$$

$$B_1 = c\big((\dot{s}_1 - \dot{s}_2)^2 - (\ddot{s}_1 - \ddot{s}_2)(s_1 - s_2)\big) - (\dot{s}_1 - \dot{s}_2)\dot{f} + f(\ddot{s}_1 - \ddot{s}_2) \tag{6.39b}$$

$$B_2 = d\big(-(\dot{s}_1 - \dot{s}_2)^2 + (\ddot{s}_1 - \ddot{s}_2)(s_1 - s_2)\big) + f(\dot{s}_1 - \dot{s}_2) - (s_1 - s_2)\dot{f}. \tag{6.39c}$$

The component is globally identifiable.

We will now consider a connection of the components and examine if it is identifiable.

Example 6.11: Connected components

Consider a SpringDamper component connected between a fixed point and a SlidingMass component. Figure 6.3 shows how graphical modeling of this system in Modelica would look.

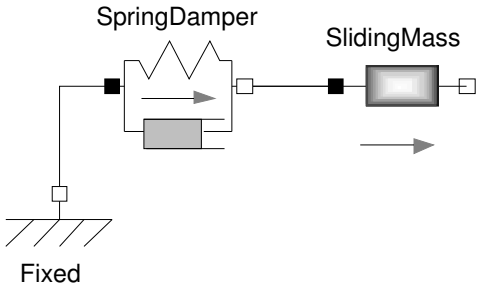


Figure 6.3: *A SpringDamper and SlidingMass connected in Modelica.*

The A polynomials of the components are

$$A_1 = \ddot{f}(\dot{s}_1 - \dot{s}_2)^2 - (\dot{s}_1 - \dot{s}_2)\dot{f}(\ddot{s}_1 - \ddot{s}_2) - (\dot{s}_1 - \dot{s}_2)f(s_1^{(3)} - s_2^{(3)}) + \dot{f}(s_1^{(3)} - s_2^{(3)})(s_1 - s_2) + f(\ddot{s}_1 - \ddot{s}_2)^2 - (\ddot{s}_1 - \ddot{s}_2)(s_1 - s_2)\ddot{f} \tag{6.40a}$$

$$A_2 = (f_1 + f_2)s^{(3)} - (\dot{f}_1 + \dot{f}_2)s^{(2)}. \tag{6.40b}$$

The components are connected so that the position s of the SlidingMass is equal to the second position s_2 of the SpringDamper and the force of the SpringDamper f is equal to the force f_1 of the SlidingMass. Furthermore, the SpringDamper is connected to a fixed point so that $s_1 = 0$, the force f_2 of the SlidingMass is controlled by the signal $u(t)$ and the position $s(t)$ is measured to give $y(t)$. This gives the connections

$$\underbrace{\begin{pmatrix} s_1 \\ s_2 - s \\ f - f_1 \\ f_2 - u \end{pmatrix}}_{g} = 0 \tag{6.41}$$

and the measurement

$$y = s. \tag{6.42}$$

Since this means that all signals but f (or equivalently f_1) are known, we need to examine if this signal is solvable from (6.40). Adding the derivative of (6.40b) to the equations (6.40) gives the following system of equations for f and its derivatives (where we have used $f = f_1$, $s_1 = 0$, and $s_2 = s$).

$$\begin{pmatrix} -\dot{s}s^{(3)} + \ddot{s}^2 & -\dot{s}\ddot{s} + s^{(3)}s & \dot{s}^2 - \ddot{s}s \\ s^{(3)} & -s^{(2)} & 0 \\ s^{(4)} & 0 & -s^{(2)} \end{pmatrix} \begin{pmatrix} f \\ \dot{f} \\ \ddot{f} \end{pmatrix} = \begin{pmatrix} 0 \\ -f_2 s^{(3)} + \dot{f}_2 s^{(2)} \\ -\dot{f}_2 s^{(3)} + \ddot{f}_2 s^{(2)} - f_2 s^{(4)} + \dot{f}_2 s^{(3)} \end{pmatrix} \tag{6.43}$$

The 3×3 matrix is invertible for most values of s and its derivatives, so the system is globally identifiable.

Note that we choose to differentiate (6.40b) since that did not introduce any new unknown variables.

6.6 Conclusions

The main conclusion that can be drawn from the discussions in this chapter is that identifiability for a component-based model can be examined using parameter-free equations. If all components are identifiable and have independent parameters, identifiability is completely determined by the parameter-free equations A, the connector equations g and the measurement equation $y = h$.

While this result is of interest in itself, an application of it to simplify examination of identifiability in component-based models. For components in model libraries, the transformation to the form (6.7) is computed once and for all and stored with the component. This makes it possible to only consider a smaller number of equations when examining identifiability for a component-based model composed of such components. Although the method described in this chapter may suffer from high computational complexity (depending, among other things, on the method selected for deciding the number of solutions for (6.22)), it can make the situation much better than when trying to use the differential-algebra approach described by Ljung and Glad (1994) directly on a complete model.

Future work could include to examine if it is possible to make the method fully automatic, so that it can be included in modeling tools and to examine if other system analysis or design methods can benefit from the modularized structure in component-based models. It could also be interesting to examine the case when several components share the same parameter. This could occur for example if the different parts of the system are affected by environmental parameters such as temperature and fluid constants.

7

Simulation-Based Tests for Identifiability

In this chapter we discuss how DAE solvers can be used for examining identifiability. The basic idea is the same as in the two previous chapters — extend the DAE with the equation $\dot{\theta}(t) = 0$ and examine if the extended DAE is observable. In this chapter, this is examined using DAE solvers.

7.1 Introduction

The development of object-oriented modeling languages such as Modelica has led to the development of effective solvers for large nonlinear DAE systems. These solvers were discussed in Section 2.5. The solvers are of course mainly intended for simulation of models, but in this chapter we will discuss how they also can be used to examine identifiability of DAE models. The method discussed here is inspired by the differential algebra approach discussed in Section 3.4.

The basic principles behind the method presented here are shown in the following example.

Example 7.1: Introductory example

Consider again the model structure (3.25a)

$$\ddot{y}(t) + 2\theta_0\dot{y}(t) + \theta_0^2 y(t) = 0 \tag{7.1}$$

which was examined by Ljung and Glad (1994). This model structure is globally identifiable, which was proved by Ljung and Glad (1994). Here it will be shown how a DAE solver can be used to prove local identifiability. If (7.1) is identifiable, it should be possible to compute the value of θ_0 given measurements of y generated from (7.1), in other

words the system of equations

$$\ddot{y}(t) + 2\theta_0\dot{y}(t) + \theta_0^2 y(t) = 0 \tag{7.2a}$$

$$\ddot{y}(t) + 2\theta(t)\dot{y}(t) + \theta^2(t)y(t) = 0 \tag{7.2b}$$

should be uniquely solvable for $\theta(t)$. This means that given a value of θ_0 and initial conditions for $y(t)$ and its derivatives, a DAE solver should be able to compute $\theta(t)$, and we should have $\theta(t) \equiv \theta_0$. The solver also computes $y(t)$ from (7.2a), but this variable is not of interest when examining identifiability.

Simulating (7.2) using Dymola for $\theta_0 = 3$, $y(0) = 0$, and $\dot{y}(0) = 1$ gives the solution $\theta(t) \equiv 3$, so the model structure is locally identifiable at 3. Simulations with other values of θ_0 give corresponding results.

The idea behind the method is thus to examine if the identification problem has several solutions by solving certain equations with a DAE solver. This chapter discusses which equations that should be solved to examine identifiability and how the results should be interpreted.

7.2 Basic Setup

In this chapter, identifiability will be discussed for general nonlinear DAE systems,

$$F\big(\dot{x}(t), x(t), \theta, u(t)\big) = 0 \tag{7.3a}$$

$$y(t) = h\big(x(t), \theta\big). \tag{7.3b}$$

However, the exact number of equations in the system is essential for the discussion. Therefore we will consider (7.3) as a set of scalar equations,

$$g_i(u, y, x, \dot{x}, \theta) = 0 \quad i = 1, 2, \ldots, r. \tag{7.4}$$

The dimensions of the variables are

$$\dim u = n_u \tag{7.5a}$$

$$\dim y = n_y \tag{7.5b}$$

$$\dim x = n_x \tag{7.5c}$$

$$\dim \theta = n_\theta. \tag{7.5d}$$

Also,

$$n_{\dot{x}} = \text{the number of } x \text{ that appear differentiated.} \tag{7.5e}$$

It is assumed that

$$r = n_y + n_x \tag{7.6}$$

so that there are the same number of equations (g_i) as unknowns (x and y) if u and θ are given. Furthermore it is assumed that the functions g_i can be differentiated with respect to time as many times as necessary.

In the system identification problem, it is known that the parameters are constant, i.e.,

$$\dot{\theta} = 0. \tag{7.7}$$

The equation system (7.4) and (7.7) is generally not solvable for arbitrary signals u and y since the number of equations, $r + n_\theta = n_y + n_x + n_\theta$, is larger than the number of unknowns, $n_x + n_\theta$. This means that they cannot be plugged directly into a DAE solver to examine identifiability. However, if the signals u and y come from an identical system, with some fixed parameter values θ_0 and internal variables $x(t)$, the system (7.4) and (7.7) must have at least one solution, since one solution is $\theta \equiv \theta_0$ and $x(t)$. If the system is not identifiable, there will be more solutions than this one. The complete problem when examining identifiability at θ_0 is thus to check if the following problem has a unique solution $\theta = \theta_0$:

$$g_i(u, y, x, \dot{x}, \theta_0) = 0 \quad i = 1, 2, \ldots, r \tag{7.8a}$$

$$g_i(u, y, \tilde{x}, \dot{\tilde{x}}, \theta) = 0 \quad i = 1, 2, \ldots, r \tag{7.8b}$$

$$\dot{\theta} = 0 \tag{7.8c}$$

Here, u and θ_0 shall be given and the solver is to compute θ, x, $\tilde{x}$, and y. We still do not have the same number of equations ($2r + n_\theta = 2n_y + 2n_x + n_\theta$) as unknowns ($n_y + 2n_x + n_\theta$). This will be further discussed in the following section. Note that it is central that (7.8a) and (7.8b) have the same inputs and outputs u and y.

If there is only one solution, it must be $\theta \equiv \theta_0$ and the system is globally identifiable at θ_0. If there are a number of distinct constant solutions θ, the system is locally, but not globally identifiable at θ_0. If there are an infinite number of solutions in every neighborhood of θ_0, then the system is neither globally nor locally identifiable at θ_0.

The identifiability properties of a model structure is normally influenced by if the initial condition $x(0)$ is known or has to be estimated. The most common case in applications is perhaps that $x(0)$ is unknown, but it could be known for example if an experiment is started in an equilibrium.

7.3 Examining Identifiability

The basic idea of the method proposed here is to solve the system of differential-algebraic equations (7.8) with respect to θ, x, $\tilde{x}$, and y given u and θ_0 using a DAE solver. If there is a locally unique solution $\theta \equiv \theta_0$, the system is locally identifiable. However, the equation system has more equations than unknowns, so it cannot be directly plugged into a DAE solver. (As was discussed in Section 2.5, currently available DAE solvers require that the number of equations is the same as the number on unknowns.) To resolve this issue, a preprocessing step is added where the equations are manipulated so that the number of equations and unknown become the same.

This section describes how the equations (7.8) should be preprocessed to allow them to be solved using a DAE solver, and how conclusions about identifiability can be drawn from the solution.

7.3.1 Preprocessing

Basically there are three problems with the description (7.8) when it comes to simulation with a DAE solver:

1. The number of equations $2r+n_\theta = 2n_y+2n_x+n_\theta$ are not the same as the number of unknowns $n_y + 2n_x + n_\theta$. As was discussed in Section 2.5, currently available DAE solvers cannot handle this situation.

2. Some of the θ may be selected as states by the solver so that initial conditions must be specified. This is a problem since the goal is to compute the value of θ.

3. Some of the $\tilde{x}$ may be selected as states so that initial conditions must be specified. This is acceptable if $x(0)$ is known in the identification problem under examination, but otherwise this is undesirable.

Problem 2 is caused by the fact that derivatives of θ are included in the system of equations. To resolve problem 1 and 2, (7.8c) must be removed. At the same time new equations must be added by differentiating the given equations to make the number of unknowns the same as the number of equations. Note that the unknowns are y, x, $\tilde{x}$, and θ so the number of unknowns is initially $n_y + 2n_x + n_\theta$, and that the number of equations (excluding $\dot{\theta} = 0$ which should be removed) is $2r = 2n_y + 2n_x$. If $n_y < n_\theta$ (which is usually the case), $n_\theta - n_y$ equations plus one equation for each new variable that is introduced in the process must be added.

The transformations may introduce new solutions that were not present in (7.8). However, the only equations that have been removed are (7.8c), so if the solution satisfies $\dot{\theta} = 0$ it must be a solution of (7.8). This will be utilized later.

The case with unknown initial conditions $x(0)$ is more involved, so the cases with known and unknown initial conditions are discussed separately below.

Known Initial Conditions

If the initial conditions are known, what needs to be done is to create $n_\theta - n_y$ new equations by differentiating equations from (7.8b) with respect to time. The equations that are differentiated should be chosen among those containing θ since the knowledge that $\dot{\theta} = 0$ can be utilized here. Differentiating other equations actually does not introduce any new information to the DAE solver, since it can differentiate equations algebraically. It is preferable to differentiate several different equations containing elements of θ since higher derivatives may make the equations more difficult to handle for the DAE solver.

Unknown Initial Conditions

If the initial conditions $x(0)$ are unknown, it is not acceptable if $\tilde{x}$ are selected as states by the DAE solver. To reduce the risk of this happening, $\dot{\tilde{x}}$ are not marked as derivatives of $\tilde{x}$ for the solver, but merely as time-dependent variables. To emphasize this, we will write $\tilde{x}_p$ instead of $\dot{\tilde{x}}$. This will introduce $n_{\dot{x}}$ new variables, so more equations need to be differentiated than for known initial conditions. Also, when differentiating, e.g., an equation containing $\tilde{x}_p$ this will produce the second derivative of $\tilde{x}$, which will also be a

new variable, denoted $\tilde{x}_{pp}$. This means that more equations need to be differentiated in this case than for the case with known initial conditions.

When differentiating equations, one should select equations to be differentiated in systematic manner, so that not too many equations are differentiated. First, equations that do not introduce new variables (e.g., $\tilde{x}_{pp}$) should be differentiated. After that, groups of equations that when differentiated give more new equations than unknowns should be differentiated. In this process, equations containing θ should be selected first since the knowledge that $\dot{\theta} = 0$ can be utilized here. Also note that all derivatives of y should be considered as known since the DAE solver can compute them from (7.8a). This process will eventually give the same number of equations as unknowns.

The procedure outlined above can be formalized using *minimally structurally singular* (MSS) sets. MSS sets were introduced by Pantelides (1988), where they were used to examine which equations to differentiate to find conditions that consistent initial conditions of a DAE must satisfy. Here they will also be used to find equations to differentiate, but with a slightly different objective. A set of equations is *structurally singular* with respect to a set of variables if the number of equations are greater than the number of variables, and a set of equations is *minimally structurally singular* if it is structurally singular, and none of its proper subsets are structurally singular. MSS sets are useful since when differentiating such a set of equations, it will produce more new equations than new differentiated variables. The following property of MSS sets will be needed.

Lemma 7.1
If a set of equations is MSS with respect to the variables occurring in the equations, then the number of equations is exactly one more than the number of variables.

Proof: The number of equations must be greater than the number of variables, otherwise the set of equations would not be structurally singular. If the number of equations exceeds the number of variables by two or more, it would be possible to remove one equation and still have a set of equations that is structurally singular. Therefore, the number of equations must exceed the number of variables by exactly one. □

The following algorithm for differentiating equations can now be formulated:

1. Let E be the original set of $n_x + n_y$ equations in (7.8b).

2. Let z be a set with n_x elements, that for each element in $\tilde{x}$ contains the highest derivative that occurs in E.

3. Find a set of equations from E that is MSS with respect to the variables in z occurring in the set. This can be done, e.g., using Algorithm 3.2 in Pantelides (1988). Preferably, the MSS set should have as few equations as possible.

4. Differentiate the equations in the MSS set. According to Lemma 7.1, the number of new equations generated will exceed the number of new variables generated by one.

5. In E, replace the equations in the MSS set with their differentiated versions.

6. Repeat from 2 until the number of equations, including those in E and those that have been removed from E, equals the number of unknowns, that is the number of x including differentiated versions plus the number of θ.

The algorithm will terminate since the difference between the number of equations and number of unknowns is reduced by one each time it reaches Step 4.

7.3.2 Drawing Conclusions on Identifiability

After the preprocessing step, there are as many equations as unknowns, so the transformed equations can be plugged into a DAE solver. What should be done now is thus to simulate the transformed equations, and examine if there is a unique solution with $\theta \equiv \theta_0$. Before simulating, the input signal u, initial condition $x(0)$ and value of θ_0 should be selected to the values where identifiability should be checked. Here it can be noted that identifiability properties often are the same for most inputs, initial states and θ_0, see e.g., Ljung and Glad (1994). Furthermore, the most time consuming parts of the process are the preprocessing and index reduction step in the solver, and these do not have to be repeated when changing u, $x(0)$, and θ_0. This means that several different choices can be tested with small computational effort. After making these choices and running the DAE solver, there are basically five situations that may occur which lead to different conclusions on the identifiability. These situations are discussed below.

The Solution θ is Constant, and $\theta \equiv \theta_0$:

The only equation that was removed from the original set of equations is (7.8c). This equation is still fulfilled since θ is constant, so the present solution is a solution of the original equations (7.8). Furthermore, since the DAE solver is assumed to give an error message if a solution is not locally unique, the solution $\theta \equiv \theta_0$ is locally unique. This gives that the system is *locally identifiable at* θ_0.

The Solution θ is Constant, and $\theta \not\equiv \theta_0$:

As in the case where $\theta \equiv \theta_0$, it is clear that the present solution is a solution of the original equations (7.8) as θ is constant. Since $\theta \not\equiv \theta_0$ it is proved that there are two different values of θ that give the same input-output behavior, so the model is not globally identifiable at θ_0. However, it is locally identifiable at the constant value θ since this solution is locally unique according to the assumptions on the DAE solver.

If it is desirable to determine if the model is locally identifiable also at θ_0, one should go back and run the simulation with a different choice of u and/or $x(0)$ to see if the new solution has $\theta \equiv \theta_0$.

If the functions g_i are polynomial, the results by Ljung and Glad (1994) give that a model structure is either locally identifiable for almost all θ or for no θ. If the g_i are polynomial, it is therefore clear that the model structure is locally identifiable at almost all θ.

The Solution θ is Time-Varying:

The original set of equations included $\dot{\theta} = 0$, so any time-varying solutions must have been introduced by the preprocessing step. This a situation that does not give information identifiability. To achieve a solution with constant θ, it may in some cases be sufficient to change u and $x(0)$. If this does not produce a constant θ, it is necessary to return to the preprocessing step and differentiate a different set of equations.

The DAE Solver Indicates that Existing Solutions are not Locally Unique:

If existing solutions are not locally unique, the preprocessing step has either introduced new solutions, or the model structure is not locally identifiable at θ_0. The case with solutions that have been added in the preprocessing step can be handled by returning to the preprocessing, differentiating a different set of equations and then running the simulation again. If the cause of a non-unique solution is that the model structure is not locally identifiable at θ_0, then this can be verified by computing which parameters or functions of the parameters that are identifiable. This is discussed in Section 7.3.3.

The DAE Solver Indicates that no Solutions Exist:

This is a degenerate case since it is clear that at least one solution exists: $\tilde{x} = x$ and $\theta = \theta_0$. The reason is usually that the initial condition $x(0)$ gives problems for the solver. For example, $x(0) = 0$ may give rise to problems. If this problem occurs, run the simulation again with a new selection of $x(0)$ and/or u.

The discussion above can be summarized with the following result.

Result 7.1
Assume that if a locally unique solution to (7.8) *exists, then it is given by the DAE solver. Otherwise the user is notified that no solution exists or that existing solutions are not locally unique. Then, for the five situations that can occur when simulating the preprocessed identifiability problem, the following conclusions can be made about identifiability for the selected input u and initial state $x(0)$:*

1. *If the solution θ is constant, and $\theta \equiv \theta_0$, then the model structure is locally identifiable at θ_0.*
2. *If the solution θ is constant, and $\theta \not\equiv \theta_0$, then the model structure is locally identifiable at θ and not globally identifiable at θ_0.*
3. *If the solution θ is time-varying, then the current solution has been introduced by the preprocessing step.*
4. *If the DAE solver indicates that existing solutions are not locally unique, then the model structure is either not locally identifiable at θ_0 or new solutions have been introduced by the preprocessing step.*
5. *If the DAE solver indicates that no solutions exist, numerical problems have occurred.*

In case 4, further examination of the identifiability properties should be done using Result 7.2 discussed below.

7.3.3 Identifiable Functions of Parameters

If a model structure is not locally identifiable, it may be interesting to examine if some parameters or functions of parameters are identifiable. This makes it possible to prove that a model structure is not locally identifiable at a certain parameter value. It may also be interesting to know which parameters that are identifiable if they represent physical quantities.

The basic observation that is used to find identifiable functions of parameters is that if a model structure is not identifiable, then it should be possible to make it identifiable by fixing one or more parameters. If for example $a + b$ is identifiable but not a and b, then b can be made identifiable by fixing a to, e.g., $a = 1$. As it is not possible to know beforehand which parameters that can be fixed, all parameters and combinations of parameters have to be tried until the model structure becomes identifiable. For each parameter (each element of θ) it is tested if the model becomes identifiable with this parameter fixed to the value of the corresponding element in θ_0. This is checked by restarting the procedure from the preprocessing step. If it cannot be made identifiable by fixing one parameter, then combinations of two parameters are tested, then combinations of three parameters, and so on.

When a parameter, or combination of parameters, that when fixed makes the model structure identifiable has been found, it is still necessary to show that the model actually is not identifiable. This is because the extra solutions that were reported by the DAE solver may be a result of the preprocessing step. To prove this, the value of the parameters are changed from their corresponding values in θ_0. If the simulation procedure still gives constant values for all parameters, it has been proven that the model structure is not globally identifiable since there are more than one set of parameter values that give the same input-output behavior. Local identifiability can be tested by making small changes in the parameters.

When some parameters that can be fixed have been found, it may be interesting to examine how the values of these parameters affect the value of the other parameters. This can be done by varying the fixed parameters to different fixed values, and noting the values of the other parameters. In this way it is possible to determine which function of the parameters that is identifiable.

This discussion leads to the following result.

Result 7.2
Consider a modified identifiability problem with the parameter vector divided into two parts (θ_1 and θ_2) where the first part θ_1 is considered known,

$$g_i(u, y, x, \dot{x}, \theta_{0,1}, \theta_{0,2}) = 0 \quad i = 1, 2, \ldots, r \tag{7.9a}$$

$$g_i(u, y, \tilde{x}, \dot{\tilde{x}}, \theta_1, \theta_2) = 0 \quad i = 1, 2, \ldots, r \tag{7.9b}$$

$$\dot{\theta}_2 = 0. \tag{7.9c}$$

If simulation of the preprocessed version of this problem with $\theta_1 \neq \theta_{0,1}$ gives a constant θ_2, then the problem is not globally identifiable at θ_0. Furthermore, the identifiable functions of the parameter are defined by $\big(\theta_2 - f(\theta_1)\big)$ where the function $f(\theta_1)$ is defined as the value of θ_2 when simulating the preprocessed version of (7.9) *for a certain θ_1.*

When using this method, it is as discussed above first necessary to find the parameters that can be varied and thus should be included in θ_1. This is done by first trying each parameter, then combinations of two parameters and so on. When a set of parameters θ_1 that makes the model identifiable when fixed have been found, the identifiable functions of the parameters are computed by changing the value of θ_1 and noting the value of θ_2.

7.4 Example

In this section, the identifiability checking procedure is exemplified on a compartmental model.

Example 7.2: Compartmental model

In this example the following model structure from Ljung and Glad (1994) is studied:

$$\dot{x}(t) = -\frac{V_m x(t)}{k_m + x(t)} - k_{01} x(t) \tag{7.10a}$$

$$x(0) = D \tag{7.10b}$$

$$y(t) = c x(t) \tag{7.10c}$$

Let the initial condition D be known, so that the unknown parameters are

$$\theta = \begin{pmatrix} V_m \\ k_m \\ k_{01} \\ c \end{pmatrix}. \tag{7.11}$$

Assume that identifiability is to be tested at

$$\theta_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}. \tag{7.12}$$

The basic setup to be simulated (7.8) is then

$$\dot{x}(t) = -\frac{1 \cdot x(t)}{2 + x(t)} - 3 \cdot x(t) \tag{7.13a}$$

$$y(t) = 4 \cdot x(t) \tag{7.13b}$$

$$\dot{\tilde{x}}(t) = -\frac{V_m(t)\tilde{x}(t)}{k_m(t) + \tilde{x}(t)} - k_{01}(t)\tilde{x}(t) \tag{7.13c}$$

$$y(t) = c(t) \cdot \tilde{x}(t) \tag{7.13d}$$

$$\theta(t) = \begin{pmatrix} V_m(t) & k_m(t) & k_{01}(t) & c(t) \end{pmatrix}^T \tag{7.13e}$$

$$\dot{\theta}(t) = 0 \tag{7.13f}$$

In the preprocessing step, (7.13f) should be removed, and $n_\theta - n_y = 3$ new equations should be added by differentiating equations. Here, (7.13c) is differentiated twice and

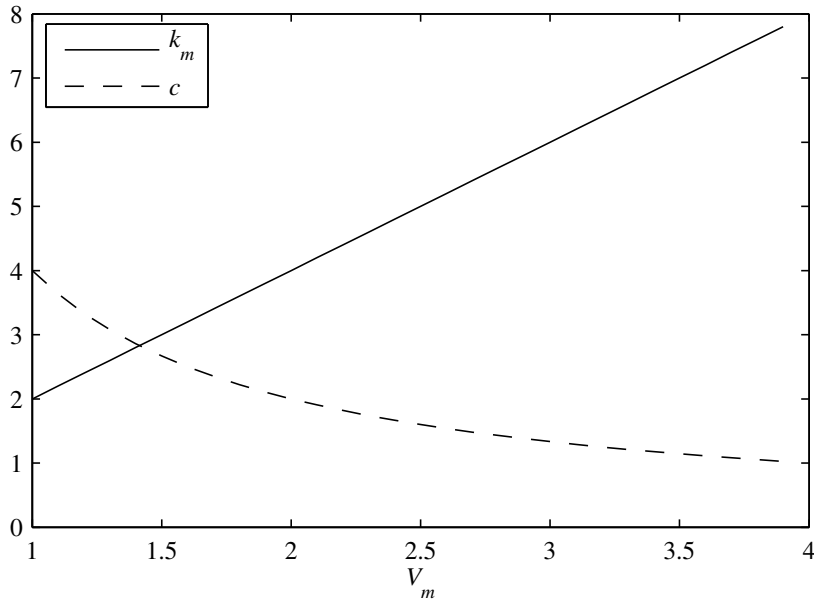


Figure 7.1: *Identifiable functions of parameters.*

(7.13d) is differentiated once to get three new equations. (7.13c) is chosen to be differentiated twice since it contains several parameters. The initial value is set to $x(0) = \tilde{x}(0) = 1$. Simulating the new system using the DAE solver in Dymola gives

$$\theta(t) \equiv \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}, \tag{7.14}$$

so the model structure is locally identifiable at this parameter value.

On the contrary to what was done above, assume now that identifiability should be examined for the case when the initial condition $x(0)$ is unknown. In this special case it is possible to see that the model structure is not identifiable without going through the preprocessing step again. This is done by simulating the same system as above but with $x(0) \neq \tilde{x}(0)$. Doing this gives a constant $\theta(t)$, but with $\theta(t) \neq \theta_0$. This directly shows that the model structure is not identifiable with unknown initial conditions. To examine which functions of the parameters that are identifiable, several different values of $\tilde{x}(0)$ are tried, and for each case the value of $\theta(t)$ is noted. During this procedure $k_{01}(t)$ is always at its true value, $k_{01}(t) = 3$, so this parameter is identifiable. The other parameters vary when $\tilde{x}(0)$ is varied, so they are not identifiable. To illustrate which functions of the parameters that are identifiable, k_m and c are plotted against V_m in Figure 7.1. The figure suggests that V_m/k_m and $V_m \cdot c$ are identifiable.

7.5 Conclusions and Ideas For Extensions

In this chapter we have discussed the possibility to examine identifiability of parameterized models. The basic idea is to simulate the system with the parameters as unknown variables and examine if there is more than one solution. However, available DAE solvers typically cannot handle the posed problems directly. Because of this a preprocessing step was discussed.

The preprocessing step, as it is described here, is heuristic and may require manual intervention. Further research efforts could thus be put at making this process fully automatic.

An interesting aspect of the method in this chapter is that DAE solvers make it possible to draw certain conclusions about dynamic systems. This could be applied to other areas than identifiability. Some possibilities are discussed below.

7.5.1 Initialization for Identification

For nonlinear models and linear models where the parameters enter nonlinearly, the system identification problem usually has to be solved as a non-convex optimization problem. This means that it is important to have a good initial guess for the parameters to avoid local minima. Perhaps the method described in this chapter could be used to solve this problem by replacing (7.8a) with measured u and y or a black-box model, similarly to what was done by Parrilo and Ljung (2003).

7.5.2 Non-Minimum Phase Systems

Linear non-minimum phase systems are characterized by, among other things, that the transfer function from output to input is unstable. This can easily be checked with a DAE solver by inverting the model (making outputs inputs and vice versa). Inverting the model is easy with a DAE solver since the model must not be written in, e.g., state-space form. All that has to be done is to specify the (original) output as a function of time, simulate the system and observe the resulting (original) input. This procedure also works for nonlinear systems.

7.5.3 Trajectory Generation

For robotic applications, trajectory generation is a common problem. Trajectory generation basically means that a control input is calculated from a desired path by inversion of a model of the robot. The inversion is usually simple to perform when the dynamics of the model are not too complicated. However, for complicated models one method to invert the model could be to use a DAE solver.

7.5.4 Observability

The problem of nonlinear observability is to examine if the internal variables, e.g., $x(t)$, of a system can be computed given measurements of the inputs and outputs. This problem is similar to the identifiability problem, and could thus also be possible to examine using

a DAE solver. Assume that the model $F(\dot{x}(t), x(t), y(t), u(t)) = 0$ should be checked for observability. This can be formulated as examining if the following DAE is uniquely solvable for unknown initial conditions on $\tilde{x}(t)$:

$$F(\dot{x}(t), x(t), y(t), u(t)) = 0 \tag{7.15a}$$

$$F(\dot{\tilde{x}}(t), \tilde{x}(t), y(t), u(t)) = 0 \tag{7.15b}$$

$$x(0) = x_0 \tag{7.15c}$$

The preprocessing step would here be somewhat different than it was for the identifiability case.

Part II

Linear DAE Models

8

Linear SDAE Models

In this chapter we discuss noise modeling in linear DAE systems.

8.1 Introduction

In Chapter 4, we discussed how noise models can be included in nonlinear DAE systems, and how this can be used to estimate internal variables and unknown parameters. In the case of linear DAE models, a more thorough analysis can be performed than for the nonlinear case, and simpler conditions for well-posedness can be derived. In the present chapter we will therefore discuss conditions that need to be satisfied to make it possible to interpret a linear SDAE with white noise inputs as an SDE, and in the following two chapters we will discuss conditions for well-posedness of parameter estimation and state estimation problems.

For continuous-time linear state-space models, a noise model can be added according to

$$\dot{x}(t) = Ax(t) + B_1u(t) + B_2v_1(t) \tag{8.1a}$$

$$y(t) = Cx(t) + v_2(t), \tag{8.1b}$$

where $v_1(t)$ and $v_2(t)$ are white noise signals. As discussed in Section 2.7 this description should be interpreted as a stochastic integral. To point this out, the notation

$$dx = Axdt + B_1u(t)dt + B_2dv_1 \tag{8.2a}$$

$$dy = Cxdt + dv_2 \tag{8.2b}$$

can be used. A Kalman filter (Anderson and Moore, 1979; Kailath et al., 2000) can then be implemented to estimate the state and predict future state values and outputs. We will here discuss what measures that need to be taken to use similar methods for linear DAE

systems. As discussed previously, it is possible to transform a linear DAE into state-space form, so it will in principle be possible to use the same methods as for state-space systems. However, there are issues on the well-posedness of DAE systems with noise models. This will be discussed below.

8.2 Noise Modeling

The natural approach to add noise to a linear DAE model is of course according to the linear SDAE shown on page 13,

$$E\dot{x}(t) = Jx(t) + K_1 u(t) + K_2 v_1(t) \tag{8.3a}$$

$$y(t) = Lx(t) + v_2(t), \tag{8.3b}$$

where $v_1(t)$ represents the unmeasured inputs and and $v_2(t)$ represents the measurement noise. K_2 is a constant matrix. This is analogous to how noise is added in a state-space model, see (8.1). It can be realized from the discussion in Section 2.3, that the internal variables $x(t)$ can depend on *derivatives* of $v_1(t)$. But if $v_1(t)$ is white noise, the derivative is not well-defined (see Section 2.7), so then the internal variables cannot depend on derivatives of $v_1(t)$. To see how this can happen, consider the following example:

Example 8.1: Linear SDAE

Consider the linear SDAE

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix} v_1(t) \tag{8.4a}$$

$$y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t). \tag{8.4b}$$

The first equation states that

$$x_1(t) = v_1(t) \tag{8.5}$$

which inserted into the second equation gives

$$x_2(t) = \dot{x}_1(t) = \dot{v}_1(t). \tag{8.6}$$

If $v_1(t)$ is white noise, this is questionable since the derivative of white noise is not well-defined. Furthermore, (8.5) is also questionable if $x_1(t)$ is a physical variable since a time-continuous white noise process has infinite variance.

In this section we derive conditions on the matrix K_2 which guarantee that $x(t)$ does not depend on derivatives of $v_1(t)$. Two equivalent conditions are derived, one using time domain methods (Section 8.2.1) and one using frequency domain methods (Section 8.2.2).

The condition that $x(t)$ does not depend on derivatives can be seen as a basic required condition on SDAE models to make it possible to interpret them as an SDE, since the derivative of white noise is not well-defined. To estimate the unknown parameters and the internal variables, further conditions must be imposed on the model. This will be discussed in the following chapters.

8.2.1 Time Domain Derivation

In this section we use time domain methods to derive a condition on K_2 which is equivalent to that derivatives of $v_1(t)$ do not affect $x(t)$.

Consider (8.3). We can rewrite the equations as

$$E\dot{x}(t) = Jx(t) + \begin{pmatrix} K_1 & K_2 \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.7a}$$

$$y(t) = Lx(t) + v_2(t). \tag{8.7b}$$

If we now consider the vector

$$\begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.8}$$

as the input and assume that the system is regular, we know from Lemma 2.3 that there exist transformation matrices P and Q such that the transformation

$$PEQQ^{-1}\dot{x}(t) = PJQQ^{-1}x(t) + P \begin{pmatrix} K_1 & K_2 \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.9}$$

gives the system

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\dot{x}(t) = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} Q^{-1}x(t) + \begin{pmatrix} B_1 & B_2 \\ D_1 & D_2 \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.10}$$

where N is a nilpotent matrix. Furthermore, Theorem 2.3 gives that the solution can be described by

$$\dot{x}_1(t) = Ax_1(t) + B_1u(t) + B_2v_1(t) \tag{8.11a}$$

$$\begin{aligned} x_2(t) &= -D_1u(t) - D_2v_1(t) \\ &\quad - \sum_{i=1}^{m-1} N^i D_1 u^{(i)}(t) - \sum_{i=1}^{m-1} N^i D_2 v_1^{(i)}(t) \end{aligned} \tag{8.11b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{8.11c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t). \tag{8.11d}$$

When we have a state-space description, $v_1(t)$ and $v_2(t)$ are white noise signals. If they were not white noise, we would technically not have a state-space description since future noise values then would depend on the current noise value. To be able to transform (8.3) into state-space form we would like to allow that $v_1(t)$ and $v_2(t)$ are white noise also here. As discussed in Section 2.7, time-continuous white noise signals require careful treatment. Most importantly, we cannot allow that any derivatives of $v_1(t)$ occur in (8.11). If $m = 1$ this requirement is trivially fulfilled and (8.11) is equivalent to the

state-space description

$$\dot{x}_1(t) = \underbrace{A}_{\tilde{A}} x_1(t) + \underbrace{B_1}_{\tilde{B}_1} u(t) + \underbrace{B_2}_{\tilde{B}_2} v_1(t) \tag{8.12a}$$

$$y(t) = \underbrace{LQ\begin{pmatrix} I \\ 0 \end{pmatrix}}_{\tilde{C}} x_1(t) + \underbrace{LQ\begin{pmatrix} 0 \\ -D_1 \end{pmatrix}}_{\tilde{D}} u(t) + \underbrace{LQ\begin{pmatrix} 0 \\ -D_2 \end{pmatrix}}_{\tilde{N}} v_1(t) + v_2(t). \tag{8.12b}$$

However, if $m > 1$, 8.11b gives that we have to require

$$ND_2 = 0 \tag{8.13}$$

to avoid differentiation of $v_1(t)$.

Note that (8.13) is related to the *impulse controllability* with respect to $v_1(t)$, see for example the book by Dai (1989b) or the original paper by Cobb (1984). If the system were impulse controllable with respect to $v_1(t)$, as many derivatives of it as possible would be included. What we need is actually the opposite of impulse controllability with respect to $v_1(t)$.

The requirement (8.13) may seem difficult to check in the original model (8.3), but in the following theorem we show that it is equivalent to the matrix K_2 being in the range of a certain matrix. This makes it possible to avoid derivatives of the noise already at the modeling stage. To formulate the theorem, we need to consider the transformation (8.9) with matrices P and Q which gives a system in the form (8.10). Let the matrix N have the singular value decomposition

$$N = U\begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} V^T = U\begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1 & V_2 \end{pmatrix}^T, \tag{8.14}$$

where V_2 contains the last k columns of V having zero singular values. Finally, define the matrix M as

$$M = P^{-1}\begin{pmatrix} I & 0 \\ 0 & V_2 \end{pmatrix}. \tag{8.15}$$

It is now possible to derive a condition on K_2.

Theorem 8.1
The condition (8.13) *is equivalent to*

$$K_2 \in \mathcal{V}(M) \tag{8.16}$$

where $\mathcal{V}(M)$ denotes the range of the matrix M, K_2 is defined in (8.3) *and M is defined in* (8.15).

The expression (8.16) means that K_2 is in the *range* of M, that is the columns of K_2 are linear combinations of the columns of M.

Proof: From Lemma 2.3 we know that there exist matrices P and Q such that

$$PEQQ^{-1}\dot{x}(t) = PJQQ^{-1}x(t) + P\begin{pmatrix} K_1 & K_2 \end{pmatrix}\begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.17}$$

gives the canonical form

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\dot{x}(t) = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} Q^{-1}x(t) + \begin{pmatrix} B_1 & B_2 \\ D_1 & D_2 \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix}. \tag{8.18}$$

Note that K_2 can be written as

$$K_2 = P^{-1} \begin{pmatrix} B_2 \\ D_2 \end{pmatrix}. \tag{8.19}$$

Let the matrix N have the singular value decomposition

$$N = U \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} V^T \tag{8.20}$$

where Σ is a diagonal matrix with nonzero elements. Since N is nilpotent it is also singular, so k singular values are zero. Partition V as

$$V = \begin{pmatrix} V_1 & V_2 \end{pmatrix}, \tag{8.21}$$

where V_2 contains the last k columns of V having zero singular values. Then $NV_2 = 0$.

We first prove the implication (8.16) $\Rightarrow$ (8.13): Assume that (8.16) is fulfilled. K_2 can then be written as

$$K_2 = M \begin{pmatrix} S \\ T \end{pmatrix} = P^{-1} \begin{pmatrix} I & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} S \\ T \end{pmatrix} = P^{-1} \begin{pmatrix} S \\ V_2 T \end{pmatrix} \tag{8.22}$$

for some matrices S and T. Comparing with (8.19), we see that $B_2 = S$ and $D_2 = V_2 T$. This gives

$$ND_2 = NV_2T = 0 \tag{8.23}$$

so (8.13) is fulfilled.

Now the implication (8.13) $\Rightarrow$ (8.16) is proved: Assume that (8.13) is fulfilled. We then get

$$0 = ND_2 = U \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} D_2 = U \begin{pmatrix} \Sigma V_1^T D_2 \\ 0 \end{pmatrix}. \tag{8.24}$$

This gives that

$$V_1^T D_2 = 0, \tag{8.25}$$

so the columns of D_2 are orthogonal to the columns of V_1, and D_2 can be written as

$$D_2 = V_2 T \tag{8.26}$$

for some matrix T. Equation (8.19) now gives

$$K_2 = P^{-1} \begin{pmatrix} B_2 \\ D_2 \end{pmatrix} = P^{-1} \begin{pmatrix} B_2 \\ V_2 T \end{pmatrix} = P^{-1} \begin{pmatrix} I & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} B_2 \\ T \end{pmatrix} = M \begin{pmatrix} B_2 \\ T \end{pmatrix} \in \mathcal{V}(M). \tag{8.27}$$

(8.16) is fulfilled.

□

We now consider how an SDAE can be transformed into state-space form. If it is assumed that the matrix K_2 in (8.3) is such that (8.13), or equivalently (8.16), is fulfilled, the form (8.11) can be written as

$$\dot{x}_1(t) = Ax_1(t) + B_1u(t) + B_2v_1(t) \tag{8.28a}$$

$$x_2(t) = -D_1u(t) - D_2v_1(t) - \sum_{i=1}^{m-1} N^iD_1u^{(i)}(t) \tag{8.28b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{8.28c}$$

$$y(t) = LQ\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t). \tag{8.28d}$$

We now proceed to transform (8.28) into a state-space description with $u^{(m-1)}(t)$ as the input using the same method as in Section 2.3.5. We thus define $x_3(t)$ according to (2.125), which gives the description

$$\dot{x}_1(t) = Ax_1(t) + \begin{pmatrix} B_1 & 0 & \dots & 0 \end{pmatrix} x_3(t) + B_2v_1(t) \tag{8.29a}$$

$$\begin{aligned} x_2(t) = &- \begin{pmatrix} D_1 & ND_1 & \dots & N^{m-2}D_1 \end{pmatrix} x_3(t) \\ &- N^{m-1}D_1u^{(m-1)}(t) - D_2v_1(t) \end{aligned} \tag{8.29b}$$

$$\dot{x}_3(t) = \begin{pmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} x_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u^{(m-1)}(t) \tag{8.29c}$$

$$y(t) = LQ\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t). \tag{8.29d}$$

Eliminating $x_2(t)$ and stacking $x_1(t)$ and $x_3(t)$ together now gives the description

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_3(t) \end{pmatrix} = \underbrace{\begin{pmatrix} A & B_1 & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{pmatrix}}_{\tilde{B}_1} u^{(m-1)}(t) + \underbrace{\begin{pmatrix} B_2 \\ 0 \end{pmatrix}}_{\tilde{B}_2} v_1 \tag{8.30a}$$

$$\begin{aligned} y(t) = &\underbrace{LQ\begin{pmatrix} I & 0 & 0 & \dots & 0 \\ 0 & -D_1 & -ND_1 & \dots & -N^{m-2}D_1 \end{pmatrix}}_{\tilde{C}} \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} + \\ &\underbrace{LQ\begin{pmatrix} 0 \\ -N^{m-1}D_1 \end{pmatrix}}_{\tilde{D}} u^{(m-1)}(t) + \underbrace{LQ\begin{pmatrix} 0 \\ -D_2 \end{pmatrix}}_{\tilde{N}} v_1(t) + v_2(t). \end{aligned} \tag{8.30b}$$

Defining

$$z(t) = \begin{pmatrix} x_1(t) \\ x_3(t) \end{pmatrix} \tag{8.31}$$

gives the more compact notation

$$\dot{z}(t) = \tilde{A}z(t) + \tilde{B}_1 u^{(m-1)}(t) + \tilde{B}_2 v_1(t) \tag{8.32a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}u^{(m-1)}(t) + \begin{pmatrix} \tilde{N} & I \end{pmatrix} \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix}. \tag{8.32b}$$

If v_1 and v_2 are white noise signals, then this description should be interpreted as a stochastic integral. To point this out, the notation

$$dz = \tilde{A}zdt + \tilde{B}_1 u^{(m-1)}dt + \tilde{B}_2 dv_1 \tag{8.33a}$$

$$dy = \tilde{C}zdt + \tilde{D}u^{(m-1)}dt + \begin{pmatrix} \tilde{N} & I \end{pmatrix} \begin{pmatrix} dv_1 \\ dv_2 \end{pmatrix} \tag{8.33b}$$

can be used. We have shown that it is possible to construct a state-space system with a noise model that describes the behavior of the linear DAE system with noise model (8.3) if $ND_2 = 0$ holds. However, the internal variables and the measured output may be white noise processes, see e.g., (8.28). This issue will be discussed in the following chapters.

Note that in the state-space model, the noise on the output equation is in general correlated with the noise on the state equation through the $v_1(t)$ term. This correlation is eliminated if $D_2 = 0$. Then $\tilde{N} = 0$ so the state-space description simplifies to

$$\dot{z}(t) = \tilde{A}z(t) + \tilde{B}_1 u^{(m-1)}(t) + \tilde{B}_2 v_1(t) \tag{8.34a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}u^{(m-1)}(t) + v_2(t). \tag{8.34b}$$

Here, the noise on the state and output equations are correlated only if $v_1(t)$ and $v_2(t)$ are.

8.2.2 Frequency Domain Derivation

In the previous section, Theorem 8.1 gave a condition on how noise can be added to a linear DAE system without making the internal variables of the system depend on derivatives of the noise. The criterion was based on a canonical form. As will be shown in this section, an equivalent result can also be derived in the frequency domain without requiring calculation of the canonical form.

Instead of examining if

$$ND_2 = 0 \tag{8.35}$$

to avoid derivatives of the noise, we will here examine if the transfer function from the process noise to the internal variables is proper (i.e., does not have higher degree in the numerator than the denominator). These two conditions are equivalent, since a transfer function differentiates its input if and only if it is non-proper. Consider the linear DAE system

$$E\dot{x}(t) = Jx(t) + K_1 u(t) + K_2 v_1(t) \tag{8.36a}$$

$$y(t) = Lx(t) + v_2(t). \tag{8.36b}$$

The question is if the transfer function

$$G(s) = (sE - J)^{-1}K_2 \tag{8.37}$$

is proper. Note that we want to examine if the *internal variables* x depend on derivatives of the noise, so L is not included in the transfer function.

Throughout the section, some concepts from the theory of *matrix fraction descriptions* (MFD) will be needed. MFDs are discussed for example by Kailath (1980) and by Rugh (1996) where they are called *polynomial fraction descriptions*.

We start by defining the *row degree* of a polynomial matrix and the concept of a *row reduced* polynomial matrix according to Rugh (1996, page 308).

Definition 8.1 (Row degree). The i:th *row degree* of a polynomial matrix $P(s)$, written as $r_i[P]$, is the degree of the highest degree polynomial in the i:th row of $P(s)$.

Definition 8.2 (Row reduced). If the polynomial matrix $P(s)$ is square ($n \times n$) and nonsingular, then it is called *row reduced* if

$$\deg[\det P(s)] = r_1[P] + \cdots + r_n[P]. \tag{8.38}$$

We will also need the following theorem from Kailath (1980):

Theorem 8.2
If the $n \times n$ polynomial matrix $D(s)$ is row reduced, then $D^{-1}(s)N(s)$ is proper if and only if each row of $N(s)$ has degree less than or equal the degree of the corresponding row of $D(s)$, i.e., $r_i[N] \leq r_i[D], i = 1, \ldots, n$.

Proof: See Kailath (1980, page 385). □

We will examine if the transfer function (8.37) (which actually is a left MFD) fulfills the conditions of Theorem 8.2. According to Rugh (1996, page 308) a MFD can be converted into row reduced form by pre-multiplication of a unimodular[1] matrix $U(s)$. More specifically, with

$$D(s) = U(s)(sE - J) \tag{8.39a}$$
$$N(s) = U(s)K_2, \tag{8.39b}$$

and consequently

$$D^{-1}(s)N(s) = (sE - J)^{-1}K_2 = G(s), \tag{8.40}$$

$D(s)$ is row reduced for a certain unimodular matrix $U(s)$. $U(s)$ is not unique, it can for example be scaled by a constant. However, Theorem 8.2 shows that for each choice of $U(s)$, the transfer function $G(s)$ of the system is proper if the highest degree of the polynomials in each row in $N(s)$ is lower than or equal to the highest degree of the polynomials in the corresponding row of $D(s)$. This gives a condition on K_2 in the following way:

[1] A polynomial matrix is called unimodular if its determinant is a nonzero real number (Rugh, 1996, page 290).

Writing $U(s)$ as

$$U(s) = \sum_{i=0}^{m} U_i s^i \tag{8.41}$$

and writing the j:th row of U_i as U_{ij}, shows that the condition

$$U_{ij} K_2 = 0 \quad i > r_j[D],\ j = 1 \ldots n \tag{8.42}$$

guarantees that the transfer function $G(s)$ of the system is proper. Here, n is the size of the square matrices E and J, or equivalently the number of elements in the vector $x(t)$.

Conversely, assume that (8.42) does not hold. Then some row degree of $N(s)$ is higher than the corresponding row degree of $D(s)$, so the transfer function $G(s)$ is then according to Theorem 8.2 not proper. This discussion proves the following theorem.

Theorem 8.3
Consider the transfer function $G(s) = (sE - J)^{-1} K_2$ where the matrices E and J are $n \times n$. Let $U(s)$ be a unimodular matrix such that $D(s) = U(s)(sE - J)$ is row reduced. Write $U(s)$ as

$$U(s) = \sum_{i=0}^{m} U_i s^i \tag{8.43}$$

and let U_{ij} be the j:th row of U_i. Then $G(s)$ is proper if and only if

$$U_{ij} K_2 = 0 \quad i > r_j[D],\ j = 1, \ldots, n. \tag{8.44}$$

Note that the criterion discussed in this section requires that the MFD is transformed into row reduced form. An algorithm for finding this transformation is provided by Rugh (1996, Chapter 16).

We have now proved two theorems, one using time domain methods and one using frequency domain methods, that give conditions which are equivalent to that $v_1(t)$ is not differentiated. This means that these two conditions are equivalent as well.

8.3 Example

In this section the results of the previous section are exemplified on a simple physical DAE system. We will use Theorem 8.1 and 8.3 to examine how a noise model can be added to a system consisting of two rotating masses as shown in Figure 8.1. It will be

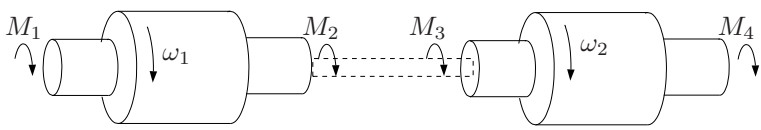


Figure 8.1: *Two interconnected rotating masses.*

shown that noise can only be added in equations where it can be physically motivated.

The system is described by the torques $M_1(t)$, $M_2(t)$, $M_3(t)$ and $M_4(t)$ and the angular velocities $\omega_1(t)$ and $\omega_2(t)$. The masses have the moments of inertia J_1 and J_2. The equations describing this system are

$$J_1\dot{\omega}_1(t) = M_1(t) + M_2(t) \tag{8.45a}$$
$$J_2\dot{\omega}_2(t) = M_3(t) + M_4(t) \tag{8.45b}$$
$$M_2(t) = -M_3(t) \tag{8.45c}$$
$$\omega_1(t) = \omega_2(t). \tag{8.45d}$$

where (8.45a) and (8.45b) describe the angular accelerations the torques produce, and (8.45c) and (8.45d) describe how the two parts are connected. Written in DAE form, these equations are

$$\begin{pmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\omega}_1(t) \\ \dot{\omega}_2(t) \\ \dot{M}_2(t) \\ \dot{M}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ -1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ M_2(t) \\ M_3(t) \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} M_1(t) \\ M_4(t) \end{pmatrix} \tag{8.46}$$

if $M_1(t)$ and $M_4(t)$ are considered as inputs. Using the transformation matrices

$$P = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ -\frac{J_2}{J_1+J_2} & \frac{J_1}{J_1+J_2} & -\frac{J_2}{J_1+J_2} & 0 \end{pmatrix} \tag{8.47}$$

$$Q = \begin{pmatrix} \frac{1}{J_1+J_2} & \frac{J_2}{J_1+J_2} & 0 & 0 \\ \frac{1}{J_1+J_2} & -\frac{J_1}{J_1+J_2} & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8.48}$$

the DAE system can be transformed into the canonical form (2.95) of Lemma 2.3. The transformation

$$z(t) = Q^{-1} \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ M_2(t) \\ M_3(t) \end{pmatrix} \tag{8.49}$$

gives

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{pmatrix} \dot{z}(t) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} z(t) + \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ -\frac{J_2}{J_1+J_2} & \frac{J_1}{J_1+J_2} \end{pmatrix} \begin{pmatrix} M_1(t) \\ M_4(t) \end{pmatrix}. \tag{8.50}$$

If we now want to incorporate noise into the DAE (8.46) by adding $K_2 v_1(t)$ to the right hand side of (8.46), which K_2-matrices are allowed? To answer this question Theorem 8.1 can be used. We begin by calculating the matrices P^{-1} and V_2 from (8.47) and (8.50). We have that

$$N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{pmatrix} \quad \Rightarrow \quad V_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{8.51}$$

and that

$$P^{-1} = \begin{pmatrix} \frac{J_1}{J_1+J_2} & 0 & 1 & -1 \\ \frac{J_2}{J_1+J_2} & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}. \tag{8.52}$$

The condition of Theorem 8.1 can now be calculated:

$$K_2 \in \mathcal{V}\left(P^{-1}\begin{pmatrix} I & 0 \\ 0 & V_2 \end{pmatrix}\right) = \mathcal{V}\left(\begin{pmatrix} \frac{J_1}{J_1+J_2} & 1 & -1 \\ \frac{J_2}{J_1+J_2} & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right) \tag{8.53}$$

This simply means that white noise cannot be added to equation (8.45d) (if $J_1 > 0$ and $J_2 > 0$). We will comment on this result below, but first we show how to derive the same condition using the frequency domain method in Theorem 8.3. Transforming the system into row reduced form gives (assuming $J_1 > 0$ and $J_2 > 0$)

$$U(s) = \begin{pmatrix} -\frac{1}{J_1} & \frac{1}{J_2} & 0 & s \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8.54}$$

$$= \underbrace{\begin{pmatrix} -\frac{1}{J_1} & \frac{1}{J_2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{U_0} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{U_1} s \tag{8.55}$$

and

$$D(s) = \begin{pmatrix} 0 & 0 & \frac{1}{J_1} & -\frac{1}{J_2} \\ 0 & J_2 s & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{pmatrix} \tag{8.56}$$

with notation from section 8.2.2.

The row degrees of $D(s)$ are $r_1[D] = 0$, $r_2[D] = 1$, $r_3[D] = 0$, and $r_4[D] = 0$. Theorem 8.3 shows that the transfer function is proper if and only if

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} K_2 = 0. \tag{8.57}$$

What equation (8.57) says is that the last row of K_2 must be zero, which is the same conclusion as was reached using the time domain method, Theorem 8.1.

The result that white noise cannot be added to the equation

$$\omega_1(t) = \omega_2(t) \tag{8.58}$$

is a result that makes physical sense since this equation represents a rigid connection. Furthermore, a noise term added to this equation would require at least one of ω_1 and ω_2 to make instantaneous changes. The equations

$$J_1\dot{\omega}_1(t) = M_1(t) + M_2(t) \tag{8.59}$$

$$J_2\dot{\omega}_2(t) = M_3(t) + M_4(t) \tag{8.60}$$

show that at least one of the torques $M_i(t)$ would have to take infinite values. This is of course not physically reasonable. Consequently, the Theorems 8.1 and 8.3 tell us how to add noise in a physically motivated way, at least for this example. They could therefore be used to guide users of object-oriented modeling software on how noise can be added to models.

8.4 Sampling with Noise Model

Also when we have a noise model, it is interesting to examine what the sampled description of a linear DAE system is. We will use Lemma 2.5 to derive the sampled counterpart of the SDAE system

$$E\dot{x}(t) = Jx(t) + K_2 v_1(t) \tag{8.61a}$$

$$y(t) = Lx(t) + v_2(t). \tag{8.61b}$$

To simplify the discussion we examine the case without input signal. A system with input signal can be handled according to what was discussed in Section 2.3.6. The noise signals $v_1(t)$ and $v_2(t)$ are interpreted as Wiener processes dv_1 and dv_2 with incremental

covariances

$$\mathrm{E}\left(dv_1\, dv_1^T\right) = Q_1 dt \tag{8.62a}$$
$$\mathrm{E}\left(dv_1\, dv_2^T\right) = Q_{12} dt \tag{8.62b}$$
$$\mathrm{E}\left(dv_2\, dv_2^T\right) = Q_2 dt. \tag{8.62c}$$

If K_2 is such that $v_1(t)$ is not differentiated, we know from Section 8.2.1 that (8.61) can be transformed into the SDE

$$dz = \tilde{A} z dt + \underbrace{\tilde{B} dv_1}_{d\tilde{v}_1(t)} \tag{8.63a}$$
$$dy = \tilde{C} z dt + \underbrace{\begin{pmatrix} \tilde{N} & I \end{pmatrix} \begin{pmatrix} dv_1 \\ dv_2 \end{pmatrix}}_{d\tilde{v}_2}. \tag{8.63b}$$

The incremental covariances of the Wiener processes $\tilde{v}_1$ and $\tilde{v}_2$ are

$$\mathrm{E}\left(d\tilde{v}_1 d\tilde{v}_1^T\right) = R_1 dt = \tilde{B} Q_1 \tilde{B}^T dt \tag{8.64a}$$
$$\mathrm{E}\left(d\tilde{v}_1 d\tilde{v}_2^T\right) = R_{12} dt = \tilde{B} \begin{pmatrix} Q_1 & Q_{12} \end{pmatrix} \begin{pmatrix} \tilde{N}^T \\ I \end{pmatrix} dt \tag{8.64b}$$
$$\mathrm{E}\left(d\tilde{v}_2 d\tilde{v}_2^T\right) = R_2 dt = \begin{pmatrix} \tilde{N} & I \end{pmatrix} \begin{pmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{pmatrix} \begin{pmatrix} \tilde{N}^T \\ I \end{pmatrix} dt. \tag{8.64c}$$

Since R_1, R_{12}, and R_2 are known for the state-space model (8.63), a sampled version of the original DAE system (8.61) can now be calculated using Lemma 2.5. We get that a sampled version of (8.61) is

$$z(T_s k + T_s) = \Phi z(T_s k) + \bar{v}(T_s k) \tag{8.65a}$$
$$y(T_s k + T_s) - y(T_s k) = \theta z(T_s k) + \bar{e}(T_s k) \tag{8.65b}$$

with

$$\Phi = e^{\tilde{A} T_s} \tag{8.66a}$$
$$\theta = \tilde{C} \int_0^{T_s} e^{\tilde{A}\tau} d\tau \tag{8.66b}$$

and

$$\mathrm{E}\left(\bar{v}(t)\bar{v}^T(t)\right) = \bar{R}_1 = \int_0^{T_s} e^{\tilde{A}(T_s-\tau)} R_1 \left(e^{\tilde{A}(T_s-\tau)}\right)^T d\tau \tag{8.67a}$$

$$\mathrm{E}\left(\bar{v}(t)\bar{e}^T(t)\right) = \bar{R}_{12} = \int_0^{T_s} e^{\tilde{A}(T_s-\tau)} \left(R_1\Theta^T(\tau) + R_{12}\right) d\tau \tag{8.67b}$$

$$\mathrm{E}\left(\bar{e}(t)\bar{e}^T(t)\right) = \bar{R}_2 = \int_0^{T_s} \Theta(\tau)R_1\Theta^T(\tau) + \Theta(\tau)R_{12} + R_{12}^T\Theta^T(\tau) + R_2\, d\tau \tag{8.67c}$$

$$\Theta(\tau) = \tilde{C}\int_\tau^{T_s} e^{\tilde{A}(s-\tau)} ds. \tag{8.67d}$$

When the measurements are sampled, it may seem awkward to first define a continuous-time measurement equation and then sample it as was proposed in this section. It is possible to instead define a discrete-time measurement equation, and this approach will be discussed in Chapters 9 and 10.

8.5 Kalman Filtering

We have now established how to transfer a linear DAE system into a discrete-time state-space system which gives an equivalent description of the output at the sampling instants. This opens up the possibility to use a discrete-time Kalman filter to estimate the states and make predictions. To be concrete, assume that we have arrived at the discrete-time state-space model

$$z(T_sk + T_s) = Az(T_sk) + Bu(T_sk) + Nv_1(T_sk) \tag{8.68a}$$

$$y(T_sk) = Cz(T_sk) + Du(T_sk) + v_2(T_sk). \tag{8.68b}$$

The implementation of a Kalman filter is then straightforward (e.g., Anderson and Moore, 1979; Kailath et al., 2000). We could also use the continuous-time state-space description (8.12) or (8.32) and implement a continuous-time Kalman filter. Note that implementation of a continuous-time Kalman filter with digital hardware always involves some sort of approximation since digital hardware operates in discrete-time.

We only get estimates of the state vector $x_1(t)$ and the output $y(t)$, not of complete vector of internal variables $x(t)$, through a normal Kalman filter. The vector $x(t)$ may not even have finite variance. This can be realized from (8.28) since $x_2(t)$ can be equal to a white noise process. In the following chapters we will therefore discuss how it can be guaranteed that all variables of interest have finite variance.

8.6 Time-Varying Linear SDAE Models

It is also interesting to examine when a time-varying linear DAE with a white noise input is well-defined so that its input-output behavior can be interpreted as a SDE. In this section

we will develop a parallel result to what was done for time-invariant linear DAE systems previously in the chapter. Consider a time-varying linear DAE as discussed in Section 2.4,

$$E(t)\dot{x}(t) = A(t)x(t) + f(t) \tag{8.69a}$$
$$y(t) = C(t)x(t). \tag{8.69b}$$

We will assume that there is a deterministic input $u(t)$ and a white noise input $v_1(t)$ so that

$$f(t) = K_1(t)u(t) + K_2(t)v_1(t). \tag{8.70}$$

There is also white measurement noise $v_2(t)$. The time-varying linear SDAE can then be written as

$$E(t)\dot{x}(t) = A(t)x(t) + K_1(t)u(t) + K_2(t)v_1(t) \tag{8.71a}$$
$$y(t) = C(t)x(t) + v_2(t). \tag{8.71b}$$

As with the time-invariant case, the problem is that derivatives of the noise process $v_1(t)$ might appear, and these are not well-defined. This can be realized from the transformations in Section 2.4 from which we get that (8.71a) is equivalent to

$$\dot{x}_1(t) = A_{13}(t)x_3(t) + f_1(t) \tag{8.72a}$$
$$0 = x_2(t) + f_2(t) \tag{8.72b}$$
$$0 = f_3(t) \tag{8.72c}$$

where

$$\begin{pmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \end{pmatrix} = P_{m+1}(t)\tilde{P}_m(t, \tfrac{d}{dt}) \cdots \tilde{P}_1(t, \tfrac{d}{dt})\big(K_1(t)u(t) + K_2(t)v_1(t)\big) \tag{8.73}$$

with P_i, $\tilde{P}_i$ defined as in Section 2.4. Since derivatives of white noise are not well-defined, $K_2(t)$ must be such that $v_1(t)$ is not differentiated. It must also be assumed that $K_1(t)u(t)$ is sufficiently differentiable. We will also assume that the DAE is regular so that x_3 and f_3 are of size zero. This is a parallel to the regularity assumption for linear time-invariant DAE systems. The conditions for when the input-output behavior of a time-varying linear SDAE is well-defined is given by the following proposition.

Proposition 8.1
Let the matrices P_i for the time-varying linear SDAE (8.69) *be defined as in Section 2.4 and the assumptions in Theorem 2.6 hold. Also assume that* (8.69) *is regular so that the size of x_3 is zero. Then the internal variables are not affected by derivatives of the noise process $v_1(t)$ if and only if*

$$\begin{pmatrix} 0 & 0 & 0 & I_{s_i} & 0 \end{pmatrix} P_i(t)P_{i-1}(t) \cdots P_1(t)K_2(t) = 0 \quad i = 1, \ldots, m, \tag{8.74}$$

that is the fourth block row of $P_i(t)P_{i-1}(t) \cdots P_1(t)K_2(t)$ is zero for $i = 1, \ldots, m$ where the division into block rows for each matrix P_i is done according to the division in Theorem 2.5.

Proof: We have that

$$\begin{pmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \end{pmatrix} = P_{m+1}(t)\tilde{P}_m(t, \tfrac{d}{dt}) \cdots \tilde{P}_1(t, \tfrac{d}{dt})\big(K_1(t)u(t) + K_2(t)v_1(t)\big) \tag{8.75}$$

where

$$\tilde{P}_i = \begin{pmatrix} I & 0 & 0 & \frac{d}{dt}I & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} P_i(t). \tag{8.76}$$

Since all the matrices P_i are invertible, (8.74) is a necessary and sufficient condition to avoid differentiation of the noise process $v_1(t)$, □

If the conditions of the theorem are satisfied we can thus write a time-varying linear SDAE as

$$\begin{pmatrix} \dot{x}_1(t) \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B_1(t, \frac{d}{dt}) \\ D_1(t, \frac{d}{dt}) \end{pmatrix} u(t) + \begin{pmatrix} B_2(t) \\ D_2(t) \end{pmatrix} v_1(t) \tag{8.77}$$

where

$$\begin{pmatrix} B_1(t, \frac{d}{dt}) \\ D_1(t, \frac{d}{dt}) \end{pmatrix} = P_{m+1}(t)\tilde{P}_m(t, \frac{d}{dt}) \cdots \tilde{P}_1(t, \frac{d}{dt})K_1(t) \tag{8.78a}$$

$$\begin{pmatrix} B_2(t) \\ D_2(t) \end{pmatrix} = P_{m+1}(t)P_m(t) \cdots P_1(t)K_2(t). \tag{8.78b}$$

This means that the input-output behavior of (8.69) can be interpreted as the SDE

$$dx_1(t) = B_1(t, \tfrac{d}{dt})u(t)dt + B_2(t)dv_1(t) \tag{8.79a}$$

$$dy(t) = C(t)x(t)dt + dv_2(t) \tag{8.79b}$$

if the conditions of Proposition 8.1 are satisfied. However, note that the internal variables $x_2(t)$ may depend directly on the noise process $v_1(t)$. This is questionable if $x_2(t)$ represent physical quantities, so in the following chapters we will discuss how this can be avoided.

8.7 Difference-Algebraic Equations

In this section, stochastic difference-algebraic equations, or stochastic discrete-time descriptor systems, are discussed.

8.7.1 Noise Modeling

A noise model can be added to a discrete-time descriptor system according to

$$Ex(t+1) = Jx(t) + K_1u(t) + K_2v_1(t) \tag{8.80a}$$

$$y(t) = Lx(t) + v_2(t), \tag{8.80b}$$

similarly to the continuous-time case. Here, $v_1(t)$ and $v_2(t)$ are uncorrelated sequences of white noise and K_2 is a constant matrix. We assume that the descriptor system is regular.

In Section 2.6 we saw that discrete-time descriptor systems may be non-causal. The stochastic system discussed here might be non-causal not only with respect to the input signal, but also with respect to the noise. This can be seen by first writing the system as

$$Ex(t+1) = Jx(t) + \begin{pmatrix} K_1 & K_2 \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \tag{8.81a}$$

$$y(t) = Lx(t) + v_2(t) \tag{8.81b}$$

and then applying Theorem 2.7. The solutions can be described by

$$x_1(t+1) = Ax_1(t) + B_1u(t) + B_2v_1(t) \tag{8.82a}$$

$$\begin{aligned} x_2(t) = &-D_1u(t) - \sum_{i=1}^{m-1} N^iD_1u(t+i) \\ &- D_2v_1(t) - \sum_{i=1}^{m-1} N^iD_2v_1(t+i) \end{aligned} \tag{8.82b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{8.82c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t). \tag{8.82d}$$

A difference from the continuous-time case is that we do not have to put any restriction on the noise model, as dependence on future values of the noise is theoretically possible. The dependence on future values of the noise can be handled for example by time shifting the noise sequence. If we define

$$\tilde{v}_1(t) = v_1(t+m-1) \tag{8.83}$$

equation (8.82) can be written as

$$x_1(t+1) = Ax_1(t) + B_1u(t) + B_2\tilde{v}_1(t-m+1) \tag{8.84a}$$

$$\begin{aligned} x_2(t) = &-D_1u(t) - \sum_{i=1}^{m-1} N^iD_1u(t+i) \\ &- D_2\tilde{v}_1(t-m+1) - \sum_{i=1}^{m-1} N^iD_2\tilde{v}_1(t+i-m+1) \end{aligned} \tag{8.84b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{8.84c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + v_2(t) \tag{8.84d}$$

which is a causal description with respect to the noise. Note that the sequences $v_1(t)$ and $\tilde{v}_1(t)$ will have the same statistical properties since they are both white noise sequences. The noise sequences $v_1(t)$ and $v_2(t)$ must be uncorrelated, otherwise $v_2(t)$ will be correlated with $\tilde{v}_1(t-m+1)$.

8.7.2 Kalman Filtering

The system (8.81) can be transformed into state-space form using the technique in Section 2.6.3. We would then get

$$z(t+1) = \tilde{A}z(t) + \tilde{B}_1 u(t+m-1) + \tilde{B}_2 v_1(t+m-1) \tag{8.85a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}_1 u(t+m-1) + \tilde{D}_2 v_1(t+m-1) + v_2(t). \tag{8.85b}$$

which, using (8.83), also can be written as

$$z(t+1) = \tilde{A}z(t) + \tilde{B}_1 u(t+m-1) + \tilde{B}_2 \tilde{v}_1(t) \tag{8.86a}$$

$$y(t) = \tilde{C}z(t) + \tilde{D}_1 u(t+m-1) + \tilde{D}_2 \tilde{v}_1(t) + v_2(t). \tag{8.86b}$$

This is a state-space description if $u(t+m-1)$ is considered as the input. However, it can be argued that dependence on future noise values is not physical, so another approach may be to require that $ND_2 = 0$, so that the system is causal with respect to the noise. We could use a similar approach as in Section 8.2 to make sure that this holds. Note again that it will not be straightforward to handle the filtering problem for (8.86) if $v_1(t)$ and $v_2(t)$ are correlated, since this would imply that $\tilde{v}_1(t-m+1)$ and $v_2(t)$ are correlated. In this case it is advisable to work with models that are causal with respect to $v_1(t)$.

When the discrete-time descriptor system has been converted into state-space form, implementation of the Kalman filter is straightforward (e.g., Anderson and Moore, 1979; Kailath et al., 2000).

Previous work on Kalman filtering of discrete-time descriptor systems is among others Deng and Liu (1999); Nikoukhah et al. (1998, 1999); Darouach et al. (1993); Dai (1987, 1989a); Chisci and Zappa (1992). The approach taken in this section for discrete-time descriptor systems, is similar to the one in Dai (1987). Dai (1987) also uses the idea to time-shift the noise sequence and write the system in state-space form, but he does not discuss how a system with input signal should be treated.

8.8 Conclusions

We noted that if noise is added to arbitrary equations of a linear DAE system, derivatives of the noise signal might affect the internal variables. Since derivatives of white noise are not well-defined, we derived a method to add noise without causing derivatives of it to affect the internal variables. Furthermore, if the SDAE system is converted into state-space form, it is possible to interpret it as an SDE and implement a Kalman filter. However, it is possible that some internal variables are equal to a white noise process, and thus have infinite variance. In the following chapters, we will discuss how this can be avoided.

We also discussed noise modeling for time-varying linear DAEs and discrete-time descriptor systems.

9

Well-Posedness of Parameter Estimation Problems

In this chapter we discuss well-posedness of parameter estimation problems for linear SDAEs, and also how the parameter estimation problems can be solved.

9.1 Introduction

In the previous chapter we discussed how noise models can be added to linear DAE systems in such a way so that the equations can be interpreted as an SDE. However, we also saw that this could lead to that some of the internal variables of the DAE have infinite variance since they are equal to a white noise process. This could possibly be accepted if the variables do not represent physical quantities, but for example sampled outputs must have finite variance. If a measured output has infinite variance, it may for example be difficult to formulate a maximum likelihood problem to estimate unknown parameters. In this chapter we will therefore discuss conditions that make the parameter estimation problem well-posed and how the parameter estimation problem can be formed. We will also discuss frequency domain methods for estimation of the parameters.

9.2 Problem Formulation

When modeling a physical system with noise, it is often reasonable that the included noise processes w_l are not white noise, but instead, for example, have a spectrum ϕ that is concentrated at low frequencies. The spectrum may also be parameterized so that it depends on the unknown parameters θ. It is also common that the initial condition is

unknown and therefore has to be parameterized. Summing up, this can be written as

$$E(\theta)\dot{x}(t) = F(\theta)x(t) + G(\theta)u(t) + \sum_{l=1}^{n_w} J_l(\theta)w_l(t,\theta) \tag{9.1a}$$

$$x(t_0,\theta) = x_0(\theta) \tag{9.1b}$$

$$\dim x(t) = n \tag{9.1c}$$

where θ is a vector of unknown parameters which lies in the domain $D_{\mathcal{M}}$ and $w_l(t,\theta)$ is a scalar Gaussian second order stationary process with spectrum

$$\phi_{w_l}(\omega,\theta). \tag{9.2}$$

The spectrum is assumed to be rational in ω with pole excess $2p_l$. This means that

$$\lim_{\omega\to\infty} \omega^{2p_l}\phi_{w_l}(\omega,\theta) = C_l(\theta)$$
$$0 < C_l(\theta) < \infty \text{ for } \theta \in D_{\mathcal{M}}.$$

It will be assumed that the input $u(t)$ is known for all $t \in [t_0, T]$ and that it is differentiable a sufficient number of times. The condition that the input is known for every t typically means that it is given at a finite number of sampling instants, and its intersample behavior between these is known, like piecewise constant, piecewise linear, or band-limited. It will be assumed the system is regular, i.e., that $\det(sE - F)$ is not zero for all s.

An output vector is measured at sampling instants t_k,

$$y(t_k) = H(\theta)x(t_k) + e(t_k) \tag{9.3}$$

where $e(t_k)$ is a Gaussian random vector with covariance matrix $R_2(k,\theta)$, such that $e(t_k)$ and $e(t_s)$ are independent for $k \neq s$ and also independent of all the processes w_l. The case with an output that is measured at discrete time instances is the most likely situation in system identification applications, so we choose to adopt this view here.

The problem treated in this chapter is to estimate the unknown parameters θ using $u(t)$ and $y(t_k)$. As mentioned earlier, problems might arise with differentiated noise or with elements of the internal variables $x(t)$ being equal to white noise (which has infinite variance). It must therefore be required that the model structure (9.1) is well-posed. The definition of well-posedness that we will use states the minimal requirements that makes it possible to form a maximum likelihood estimator for the parameters. The first requirement is that the DAE is regular, since this guarantees a unique solution in the absence of noise. The second requirement is that the sampled measurements $y(t_k)$ have finite variance. This means that the equations do not implicitly specify that $y(t_k)$ contains continuous-time white noise or derivatives of continuous-time white noise.

Definition 9.1 (Well-posedness). Let $x(t)$ be defined as the solution to (9.1) for a $\theta \in D_{\mathcal{M}}$. The problem to estimate θ from knowledge of $u(t), t \in [t_0, T]$ and $y(t_k), k = 1,\ldots,N;\ t_k \in [t_0, T]$ is *well-posed* if $H(\theta)x(t_k)$ has finite variance and (9.1) is regular for all $\theta \in D_{\mathcal{M}}$.

Note that the initial value $x_0(\theta)$ may not be chosen freely when computing $x(t,\theta)$ (see Section 9.5). The possibly conflicting values in $x_0(\theta)$ will be ignored, and actually have

no consequence for the computation of $x(t, \theta)$ for $t > t_0$. For a well-posed estimation problem the likelihood function which is the value of the joint probability density function for the random vectors $y(t_k)$ at the actual observations can be computed. Also this will be discussed in Section 9.5.

9.3 Main Result

The main result of this chapter is the characterization of a well-posed model structure, which is presented in this section. Before presenting the result, some notation must be introduced. Let the range and null space of a matrix A be denoted by

$$\mathcal{V}(A) \text{ and } \mathcal{N}(A)$$

respectively. Furthermore, the following definition of an oblique projection will be used.

Definition 9.2 (Oblique projection). Let B and C be spaces with $B \cap C = \{0\}$ that together span $\mathbb{R}^n$. Let the matrices $\bar{B}$ and $\bar{C}$ be bases for B and C respectively. The *oblique projection* of a matrix A along B on C is defined as

$$A/_B\, C \triangleq \begin{pmatrix} 0 & \bar{C} \end{pmatrix} \begin{pmatrix} \bar{B} & \bar{C} \end{pmatrix}^{-1} A. \tag{9.4}$$

Note that the projection is independent of the choice of bases for B and C. This definition basically follows the definition by van Overschee and De Moor (1996, Section 1.4.2). However, we here consider projections along column spaces instead of row spaces. Also, the conditions on the spaces B and C give a simpler definition. The more general version by van Overschee and De Moor (1996) is not necessary here. The main result can now be formulated as follows.

Theorem 9.1
Consider the model (9.1)*. Let $\lambda(\theta)$ be a scalar such that $\lambda(\theta)E(\theta) + F(\theta)$ is invertible. Let*

$$\bar{E}(\theta) = \big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1} E(\theta). \tag{9.5}$$

Assuming the model (9.1) *is regular, the estimation problem* (9.1)–(9.3) *is well-posed if and only if*

$$\Big[\bar{E}^j(\theta)\big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1} J_l(\theta)\Big] \Big/_{\mathcal{V}\left(\bar{E}^n(\theta)\right)} \mathcal{N}\big(\bar{E}^n(\theta)\big) \in \mathcal{N}\big(H(\theta)\big) \quad j \geq p_l, \forall l. \tag{9.6}$$

Proof: See Appendix B. □

Note that any $\lambda(\theta)$ can be used to check if an estimation problem is well-posed, as long as $\lambda(\theta)E(\theta) + F(\theta)$ is invertible. This follows directly from the theorem, since (9.6) is equivalent to well-posedness for every $\lambda(\theta)$ with invertible $\lambda(\theta)E(\theta) + F(\theta)$.

9.4 Measuring Signals with Infinite Variance

It may happen that a selected output has infinite instantaneous variance. This happens when condition (9.6) is violated. This is best illustrated by an example: Let the SDAE be

$$\dot{x}_1(t) = -2x_1(t) + v(t) \tag{9.7a}$$
$$0 = -x_2(t) + v(t) \tag{9.7b}$$

where $v(t)$ is continuous-time white noise. We would like to measure $x_1 + x_2$. This is not a well-posed problem since x_2 has infinite variance. A convenient way of dealing with this in a modeling situation, would be to explicitly introduce a presampling, low pass filter, to make the measured variable

$$x_3(t) = \frac{1}{0.01p + 1}\big(x_1(t) + x_2(t)\big).$$

Including this new variable in the SDAE gives

$$\begin{aligned}
\dot{x}_1(t) &= -2x_1(t) + v(t) \\
\dot{x}_3(t) &= -100x_3(t) + 100x_1(t) + 100v(t) \\
0 &= -x_2(t) + v(t)
\end{aligned}$$

with the sampled measurements

$$y(t_k) = x_3(t_k) + e(t_k).$$

This is a well-posed problem. The method suggested here is related to the sampling method described in Lemma 2.5.

9.5 The Log-Likelihood Function and the Maximum Likelihood Method

To implement the maximum likelihood method for parameter estimation, it is necessary to compute the likelihood function. The likelihood function for the estimation problem is computed from the joint probability density function of the observations $y(t_k)$. It is customary to determine this from the conditional densities $p[y(t_k)|y(t_0)\dots y(t_{k-1}), u(\cdot), \theta]$. See, e.g., Ljung (1999, Section 7.4). In other words, we need the one-step-ahead predictions of the measured outputs.

By representing the disturbances $w_l(t, \theta)$ as outputs from linear filters driven by white noise $v_l(t)$ (which is possible, since they have rational spectral densities), the SDAE can be transformed into state-space form using the techniques discussed in Section 2.3. This is done by first representing the noise processes $w_l(t, \theta)$ as

$$\dot{x}_w(t) = A_w(\theta)x_w(t) + B_w(\theta)v(t) \tag{9.8a}$$
$$w(t, \theta) = C_w(\theta)x_w(t) + D_w(\theta)v(t) \tag{9.8b}$$

where

$$v(t) = \begin{pmatrix} v_1(t) \\ \vdots \\ v_{n_v}(t) \end{pmatrix} \tag{9.9}$$

is white noise with covariance $R_1(\theta)\delta(t)$ and

$$w(t,\theta) = \begin{pmatrix} w_1(t,\theta) \\ \vdots \\ w_{n_w}(t,\theta) \end{pmatrix}. \tag{9.10}$$

As discussed in Section 2.7.2, this should be interpreted as a stochastic integral. By writing

$$J(\theta) = \begin{pmatrix} J_1(\theta) & \cdots & J_{n_w}(\theta) \end{pmatrix} \tag{9.11}$$

(9.1), (9.3), and (9.8) can be combined to give

$$\begin{aligned} \begin{pmatrix} E(\theta) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{x}_w(t) \end{pmatrix} &= \begin{pmatrix} F(\theta) & J(\theta)C_w(\theta) \\ 0 & A_w(\theta) \end{pmatrix} \begin{pmatrix} x(t) \\ x_w(t) \end{pmatrix} \\ &\quad + \begin{pmatrix} G(\theta) \\ 0 \end{pmatrix} u(t) + \begin{pmatrix} J(\theta)D_w(\theta) \\ B_w(\theta) \end{pmatrix} v(t) \end{aligned} \tag{9.12a}$$

$$y(t_k) = \begin{pmatrix} H(\theta) & 0 \end{pmatrix} \begin{pmatrix} x(t_k) \\ x_w(t_k) \end{pmatrix} + e(t_k). \tag{9.12b}$$

Under the assumption of regularity, this DAE can, using Theorem 2.3, be transformed into the form

$$\dot{x}_1(t) = A(\theta)x_1(t) + G_1(\theta)u(t) + J_1(\theta)v(t) \tag{9.13a}$$

$$x_2(t) = -\left(I + \frac{d}{dt}N(\theta) + \cdots + \frac{d^{m-1}}{dt^{m-1}}N^{m-1}(\theta)\right)\big(G_2(\theta)u(t) + J_2(\theta)v(t)\big) \tag{9.13b}$$

$$y(t_k) = C_1(\theta)x_1(t_k) + C_2(\theta)x_2(t_k) + e(t_k). \tag{9.13c}$$

Inserting (9.13b) into (9.13c) gives (omitting dependence on θ)

$$y(t_k) = C_1x_1(t_k) - C_2\sum_{l=1}^{m}\left(\frac{d^{l-1}}{dt^{l-1}}N^{l-1}\big(G_au(t_k) + J_av(t_k)\big)\right) + e(t_k).$$

If it is assumed that the SDAE forms a well-posed estimation problem, $y(t_k)$ does not depend on time-continuous white noise, i.e., $v(t)$. This means that $y(t_k)$ can be written as

$$y(t_k) = C_1(\theta)x_1(t_k) - C_2(\theta)\sum_{l=1}^{m}\left(\frac{d^{l-1}}{dt^{l-1}}N^{l-1}(\theta)G_2(\theta)u(t_k)\right) + e(t_k).$$

Summing up, the original linear SDAE can be transformed into the form

$$\dot{x}_1(t) = A(\theta)x_1(t) + G_1(\theta)u(t) + J_1(\theta)v(t) \tag{9.14a}$$

$$y(t_k) = C_1(\theta)x_1(t_k) - C_2(\theta)\sum_{l=1}^{m}\left(\frac{d^{l-1}}{dt^{l-1}}N^{l-1}(\theta)G_2(\theta)u(t_k)\right) + e(t_k) \tag{9.14b}$$

$$v(t) = \begin{pmatrix} v_1(t) & v_2(t) & \cdots & v_{n_v}(t)\end{pmatrix}^T \tag{9.14c}$$

$$Ev(t)v^T(s) = R_1(\theta)\delta(t-s) \tag{9.14d}$$

$$Ee(t_k)e^T(t_s) = R_2(k,\theta)\delta_{t_k,t_s}. \tag{9.14e}$$

This is a standard linear prediction problem with continuous-time dynamics, continuous-time white noise, and discrete-time measurements. The Kalman filter equations for this are given, e.g., by Jazwinski (1970), and they define the one-step-ahead predicted outputs $\hat{y}(t_k|t_{k-1},\theta)$ and the prediction error variances $\Lambda(t_k,\theta)$. With Gaussian disturbances, we obtain in the usual way the log-likelihood function

$$V_N(\theta) = \frac{1}{2}\sum_{k=1}^{N}\big(y(t_k) - \hat{y}(t_k|t_{k-1},\theta)\big)^T\Lambda^{-1}(t_k,\theta) \times \big(y(t_k) - \hat{y}(t_k|t_{k-1},\theta)\big) + \log\det\Lambda(t_k,\theta). \tag{9.15}$$

The parameter estimates are then computed as

$$\hat{\theta}_{ML} = \arg\min_{\theta} V_N(\theta). \tag{9.16}$$

If a general norm of the prediction errors,

$$\varepsilon(t_k,\theta) = y(t_k) - \hat{y}(t_k|t_{k-1},\theta), \tag{9.17}$$

is minimized, we get the prediction error method.

In practice, the important question of how the state-space description should be computed remains. As discussed in Chapter 11, the form (9.14) can be computed using numerical software. But if some elements of the matrices are unknown, numerical software cannot be used. Another approach could be to calculate the canonical forms using symbolical software. But this approach has not been thoroughly investigated, and symbolical software is usually not as easily available as numerical software. The remedy is to make the conversion using numerical software for each value of the parameters that the identification algorithm needs. Consider for example the case when the parameters are to be estimated by minimizing (9.15) using a Gauss-Newton search. For each parameter value θ that the Gauss-Newton algorithm needs, the transformed system (9.14) can be computed.

If the initial condition of the system is unknown, it should be estimated along with the parameters. For state-space systems, this is done by parameterizing the initial state, $x(t_0) = x_0(\theta)$. For linear SDAE systems care must be taken when parameterizing the initial value. From (B.3) on page 197 of Appendix B we get that

$$x(t_0) = \begin{pmatrix} T_1(\theta) & T_2(\theta)\end{pmatrix}\begin{pmatrix} x_s(t_0) \\ x_a(t_0)\end{pmatrix}. \tag{9.18}$$

It is also obvious from the transformed system equations (B.4a) and (B.8) that $x_s(t_0)$ can be parameterized freely, while $x_a(t_0)$ is specified by the input and noise signals. The part of $x(t_0)$ that can be parameterized is thus

$$x_s(t_0) = x(t_0)/_{\mathcal{V}(T_2)}\mathcal{V}(T_1) = x(t_0)/_{\mathcal{N}\left(\bar{E}^n(\theta)\right)}\mathcal{V}\left(\bar{E}^n(\theta)\right) \tag{9.19}$$

where $\bar{E}(\theta)$ is the matrix defined in (9.5). Note that since x_a is determined by (B.8), any initial conditions that are specified for x_a can be ignored in the identification procedure since they do not affect the likelihood function.

9.6 Frequency Domain Identification

The work which has been done this far has been based on transforming the DAE system into a state-space-like system, and using identification methods for state-space descriptions. As was discussed earlier, this transformation always exists if the system is regular, and can be computed numerically. However, we have seen that the work to transform a linear DAE system into state-space form might be significant in some cases. Furthermore, the output can depend on derivatives of the input. If the input can be selected, then it might be possible to differentiate it analytically. If, on the other hand, only a measured input is available, it must be differentiated numerically, which can be a problem if the signal is noisy.

Here, we examine another approach to the identification problem that offers an alternative way to handle these potential problems, namely identification in the frequency domain. The conversion into state-space form can be avoided in the output error case as we will see below. A model which differentiates the input will have a large amplification for high frequencies. In the frequency domain we could therefore handle this problem by not including measurements with a too high frequency in $Z^N = \{U(\omega_1), Y(\omega_1), \ldots, U(\omega_N), Y(\omega_N)\}$.

As discussed in Section 3.3, it is assumed that the model structure is specified by transfer functions (or matrices of transfer functions) according to

$$y(t) = G(p,\theta)u(t) + H(p,\theta)e(t) \tag{9.20}$$

when performing frequency domain identification. $H(p,\theta)$ is assumed to have a causal inverse.

A linear DAE system with only measurement noise (an output error model),

$$E(\theta)\dot{x}(t) = J(\theta)x(t) + K_1(\theta)u(t) \tag{9.21a}$$
$$y(t) = L(\theta)x(t) + e(t), \tag{9.21b}$$

can be transformed directly into the form (9.20) under the usual assumption of regularity. The only difference from the transfer function of a state-space system is that $G(p,\theta)$ may be non-proper here. The transfer functions are

$$G(p,\theta) = L(\theta)\left(pE(\theta) - J(\theta)\right)^{-1} K_1(\theta) \tag{9.22a}$$
$$H(p,\theta) = 1. \tag{9.22b}$$

When the transfer function has been calculated, all we have to do is to plug it into any identification algorithm for the frequency domain. Books which treat this are e.g., Ljung (1999) and Pintelon and Schoukens (2001). Note that $G(p,\theta)$ here easily could be calculated using symbolical software. We can therefore compute $G(p,\theta)$ once and for all, and do not have to perform the calculation for each parameter value. One possible selection of identification method is to minimize the criterion

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} \|Y(\omega_k) - G(i\omega_k,\theta)U(\omega_k)\|^2 \tag{9.23}$$

with respect to the parameters θ.

Estimates of the Fourier transforms of the input and output signals are needed. As discussed in Section 3.3, these could be provided directly by a special measurement device or estimated from time domain data. A drawback with identification in the frequency domain is that knowledge of the initial values of the internal variables is more difficult to utilize than for time domain identification.

In the more complex case when the model also has process noise,

$$E(\theta)\dot{x}(t) = J(\theta)x(t) + K_1(\theta)u(t) + K_2(\theta)v_1(t) \tag{9.24a}$$

$$y(t) = L(\theta)x(t) + v_2(t), \tag{9.24b}$$

the noise filter $H(p,\theta)$ cannot be calculated in a straightforward manner. One approach to calculate $H(p,\theta)$ here is to first transform the DAE system into state-space form and then compute the Kalman filter. We now in principle need to do the same transformation that needs to be done when estimating the parameters in the time domain. We therefore do not have the possibility of calculating $H(p,\theta)$ once and for all with symbolical software as could be done for the output error case.

9.7 Time-Varying Linear SDAE Models

In this section we will examine well-posedness of the problem to estimate unknown parameters θ in the time-varying linear DAE

$$E(t,\theta)\dot{x}(t) = F(t,\theta)x(t) + f(t,\theta) \tag{9.25a}$$

$$x(t_0,\theta) = x_0(\theta) \tag{9.25b}$$

$$\dim x(t) = n. \tag{9.25c}$$

where

$$f(t,\theta) = G(t,\theta)u(t) + \sum_{l=1}^{n_w} J_l(t,\theta)w_l(t,\theta). \tag{9.26}$$

Measurements are collected at time instances t_k,

$$y(t_k) = H(t_k,\theta)x(t_k) + e(t_k). \tag{9.27}$$

As before, $w_l(t,\theta)$ is a Gaussian second order stationary process with spectrum

$$\phi_{w_l}(\omega,\theta) \tag{9.28}$$

where the spectrum is assumed to be rational in ω with pole excess $2p_l$. We will use the definition of well-posedness from the time-invariant case modified for the time varying case.

Definition 9.3 (Well-posedness). Let $x(t)$ be defined as the solution to (9.25a) for a $\theta \in D_{\mathcal{M}}$. The problem to estimate θ from knowledge of $u(t), t \in [t_0, T]$, and $y(t_k), k = 1, \ldots, N;\ t_k \in [t_0, T]$ is *well-posed* if $H(t_k, \theta)x(t_k)$ has finite variance and (9.25a) is regular for all $\theta \in D_{\mathcal{M}}$.

Here, regularity means that no part of x is undetermined as discussed in Section 2.4. To examine well-posedness of (9.25), we examine (9.25a) transformed according to Theorem 2.6.

$$\dot{x}_1(t) = A_{13}(t,\theta)x_3(t) + f_1(t,\theta) \tag{9.29a}$$

$$0 = x_2(t) + f_2(t,\theta) \tag{9.29b}$$

$$0 = f_3(t,\theta) \tag{9.29c}$$

where

$$\begin{pmatrix} f_1(t,\theta) \\ f_2(t,\theta) \\ f_3(t,\theta) \end{pmatrix} = P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt}) \cdots \tilde{P}_1(t,\theta,\tfrac{d}{dt}) \times \left(G(t,\theta)u(t) + \sum_{l=1}^{n_w} J_l(t,\theta)w_l(t,\theta) \right). \tag{9.30}$$

The system is assumed to be regular, so x_3 and f_3 are of size zero. We want to examine if

$$H(t_k,\theta)x(t_k) = H(t_k,\theta)Q(t_k,\theta) \begin{pmatrix} x_1(t_k) \\ x_2(t_k) \end{pmatrix} \tag{9.31}$$

has finite variance (x_3 is removed since it is of size zero). Theorem 2.6 gives that f_1 does not depend on derivatives of f, so x_1 is always well-defined with finite variance through the SDE

$$\dot{x}_1(t) = f_1(t,\theta). \tag{9.32}$$

We must thus examine if

$$H(t_k,\theta)Q(t_k,\theta) \begin{pmatrix} 0 \\ x_2(t_k) \end{pmatrix} = H(t_k,\theta)Q(t_k,\theta) \begin{pmatrix} 0 \\ -f_2(t_k) \end{pmatrix} \tag{9.33}$$

has finite variance. For this expression to have finite variance, it must be guaranteed that it does not depend on too high derivatives of w_l, $l = 1, \ldots, n_w$. Each w_l can be differentiated at most $p_l - 1$ times since its spectrum has pole excess $2p_l$. This can be realized from (2.179b) which gives that the variance of $\frac{d^n}{dt^n} w_l$ is

$$r(0) = \int_{-\infty}^{\infty} (iw)^{2n} \phi_{w_l} d\omega \begin{cases} < \infty & \text{if } n \leq p_l - 1 \\ = \infty & \text{if } n \geq p_l. \end{cases} \tag{9.34}$$

Further transforming (9.33) we get

$$H(t_k,\theta)Q(t_k,\theta)\begin{pmatrix}0\\ x_2(t_k)\end{pmatrix} = H(t_k,\theta)Q(t_k,\theta)\begin{pmatrix}0 & 0\\ 0 & I\end{pmatrix}\begin{pmatrix}f_1(t_k,\theta)\\ f_2(t_k,\theta)\end{pmatrix} =$$
$$= H(t,\theta)Q(t,\theta)\begin{pmatrix}0 & 0\\ 0 & I\end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})\times$$
$$\left(G(t,\theta)u(t)+\sum_{l=1}^{n_w} J_l(t,\theta)w_l(t,\theta)\right)\Bigg|_{t=t_k}. \quad (9.35)$$

Note that the derivative should be applied before inserting $t = t_k$. The expression shows that we must require that the expression

$$H(t,\theta)Q(t,\theta)\begin{pmatrix}0 & 0\\ 0 & I\end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J_l(t,\theta)w_l(t,\theta)\Bigg|_{t=t_k} \quad (9.36)$$

for $l = 1,\ldots,n_w$ does not contain higher derivatives than $p_l - 1$ of w_l. We formalize this result with a proposition.

Proposition 9.1
The estimation problem (9.25)–(9.28) *is well-posed if and only if* (9.25a) *is regular and*

$$H(t,\theta)Q(t,\theta)\begin{pmatrix}0 & 0\\ 0 & I\end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J_l(t,\theta)\Bigg|_{t=t_k} \quad (9.37)$$

is of at most order $p_l - 1$ in $\frac{d}{dt}$ for $l = 1,\ldots,n_w$, for all t_k, and $\theta \in D_{\mathcal{M}}$. The matrices P and Q are the transformation matrices defined in Section 2.4.

Note that the derivatives should be handled as operators when applied to J_l, for example

$$\frac{d}{dt}t = 1 + t\frac{d}{dt}. \quad (9.38)$$

When an estimation problem is well-posed, (9.32) and (9.27) can be used to compute the likelihood function for the output.

9.8 Difference-Algebraic Equations

In this section, stochastic difference-algebraic equations, or stochastic discrete-time descriptor systems, are discussed.

9.8.1 Time Domain Identification

As discussed in Section 2.6.3 and 8.7, a discrete-time descriptor system can be transformed into a discrete-time state-space system. We can therefore use the prediction error method or maximum likelihood method as described in Chapter 3. However, as in the continuous-time case, we are faced with the choice of either calculating the state-space

description symbolically, or doing it numerically. The approach suggested here is to compute it numerically for each parameter value that a state-space description is necessary, since it is discussed in the previous chapters how this transformation can be performed.

Consider for example the case when we wish to estimate the parameters by minimizing the prediction error criterion

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\varepsilon^T(t,\theta)\Lambda^{-1}\varepsilon(t,\theta) \tag{9.39}$$

using a Gauss-Newton search. As in the continuous-time case, we for each parameter value θ that the Gauss-Newton algorithm needs compute a state-space description using the methods in Chapter 11 and then calculate the prediction errors $\varepsilon(t,\theta)$.

In can be noted that the discrete-time parameter estimation problem does not suffer from the same well-posedness issues as the continuous-time case. This is because discrete-time noise processes always have finite variance.

9.8.2 Frequency Domain Identification

Analogously to the continuous-time case, frequency domain identification is a way to avoid having to transform the descriptor system into state-space form. For frequency domain identification in discrete-time, it is assumed that the system is described by

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t), \tag{9.40}$$

as discussed in Section 3.3. $H(q,\theta)$ is assumed to have a causal inverse.

A linear discrete-time descriptor system with an output error noise model,

$$E(\theta)x(t+1) = J(\theta)x(t) + K_1(\theta)u(t) \tag{9.41a}$$
$$y(t) = L(\theta)x(t) + e(t), \tag{9.41b}$$

has the transfer functions

$$G(q,\theta) = L(\theta)\left(qE(\theta) - J(\theta)\right)^{-1}K_1(\theta) \tag{9.42a}$$
$$H(q,\theta) = 1. \tag{9.42b}$$

We can here plug $G(q,\theta)$ directly into a criterion like

$$V_N(\theta, Z^N) = \sum_{k=1}^{N}\left|Y(\omega_k) - G(e^{i\omega_k},\theta)U(\omega_k)\right|^2. \tag{9.43}$$

As in the continuous-time case, the situation is more complicated if we have a full noise model as in

$$E(\theta)x(t+1) = J(\theta)x(t) + K_1(\theta)u(t) + K_2(\theta)v_1(t) \tag{9.44a}$$
$$y(t) = L(\theta)x(t) + v_2(t). \tag{9.44b}$$

Also here, the simplest way to calculate $H(q,\theta)$ is probably to go via a state-space description. Consequently, not much is gained compared to using a time domain method.

9.9 Conclusions

The main result of this chapter is Theorem 9.1, where we provide necessary and sufficient conditions for a parameter estimation problem, formed from a linear SDAE, to be well-posed. We also discussed how the parameter estimation problem can be formed for well-posed problems, both in the time domain and in the frequency domain. Time-varying DAEs and the discrete-time case were also briefly treated.

10

Well-Posedness of State Estimation Problems

In this chapter we discuss well-posedness of state estimation problems for linear SDAEs, and also how these problems can be solved using the Kalman filter.

10.1 Introduction

In the previous chapter we discussed well-posedness of parameters estimation problems, and concluded that it must be required that the measured output has finite variance to allow maximum likelihood estimation of unknown parameters. In this chapter we will discuss state estimation problems, that is estimation of the internal variables of a linear SDAE. To allow estimation of the internal variables, it must be required that they have finite variance. We will first discuss the case when the SDAE has colored noise inputs and discrete-time measurements similarly to the case examined for parameter estimation in the previous chapter. We will then examine the case when the input is white noise and the output is not sampled, similarly to the problem solved by continuous-time Kalman filters.

For references to previous works on well-posedness of state estimation problems, see Section 4.2.

10.2 Formulations without Continuous-Time White Noise

We shall in this section give a formulation of an SDAE filtering problem that only explicitly employs stochastic variables with finite variance, similarly to what was done in the previous chapter. We shall then investigate if it corresponds to a mathematically well-

posed problem. We will therefore consider an SDAE

$$E\dot{x}(t) = Fx(t) + Gu(t) + \sum_{l=1}^{n_w} J_l w_l(t) \tag{10.1a}$$

$$x(t_0) = x_0 \tag{10.1b}$$

$$\dim x(t) = n \tag{10.1c}$$

where $w_l(t)$ is a Gaussian second order stationary process with spectrum $\phi_{w_l}(\omega)$ which is rational in ω with pole excess $2p_l$. Recall that this means that

$$0 < \lim_{\omega\to\infty} \omega^{2p_l}\phi_{w_l}(\omega) < \infty. \tag{10.1d}$$

The input $u(t)$ is known for all $t \in [t_0, T]$. It will also be assumed that it is differentiable a sufficient number of times. An output vector is measured at sampling instants t_k,

$$y(t_k) = Hx(t_k) + e(t_k), \quad k = 1, \ldots, N \tag{10.1e}$$

where $e(t_k)$ is a Gaussian random vector with covariance matrix R_k, such that $e(t_k)$ and $e(t_s)$ are independent for $k \neq s$ and also independent of all the processes $w_l(t)$.

It is a feature of the modeling techniques mentioned in the introduction that they often introduce a number of variables that only play a role in intermediate calculations and are of no interest in themselves. Therefore we introduce the variable $\bar{x}$ where all (linear combinations of) components of x that are of interest are collected,

$$\bar{x}(t) = Mx(t) \tag{10.2}$$

for some rectangular matrix M.

The estimation problem considered here is well-posed if both the variables to be estimated $\bar{x}$ and the measured output y have finite variance. This differs from the formulation in the previous chapter since we also require the internal variables to have finite variance.

Definition 10.1 (Well-posedness). Let $x(t)$ be defined as the solution to (10.1). The problem to estimate $\bar{x}(t) = Mx(t)$ from $y(t_k)$, $k = 1, \ldots, N; t_k \in [t_0, T]$ and $u(t)$, $t \in [t_0, T]$, is *well-posed* if $Hx(t_k)$ and $Mx(t)$ have finite variances and (10.1) is regular.

We shall find that a well-posed filtering problem can be solved by the regular Kalman filter.

It can be noted that the initial value $x(t_0)$ cannot be chosen freely, since part of x is determined by the deterministic input u. Only

$$x(t_0)/_{\mathcal{N}(\bar{E}^n)}\, \mathcal{V}(\bar{E}^n) \tag{10.3}$$

can be given an arbitrary value. Any conflicting values of $x(t_0)$ will be ignored and have no consequence for the estimation of $x(t)$, $t > t_0$.

The result on well-posedness of the state estimation problem is similar to the result on well-posedness of the parameter estimation problem, but it must also be required that the

internal variables of interest, $\bar{x}$, have finite variance. To formulate the result, we recall the definition of an oblique projection of a matrix A along the space B on the space C,

$$A/_B\, C \triangleq \begin{pmatrix} 0 & \bar{C} \end{pmatrix} \begin{pmatrix} \bar{B} & \bar{C} \end{pmatrix}^{-1} A \tag{10.4}$$

where $\bar{B}$ and $\bar{C}$ are bases for B and C respectively. We can now formulate the main result of this section.

Theorem 10.1
Consider (10.1). *Let λ be a scalar such that $(\lambda E + F)$ is invertible. Let*

$$\bar{E} = (\lambda E + F)^{-1} E. \tag{10.5}$$

Then the estimation problem (10.1) *is well-posed if and only if*

$$\left[\bar{E}^j (\lambda E + F)^{-1} J_l\right] /_{\mathcal{V}(\bar{E}^n)} \mathcal{N}(\bar{E}^n) \in \mathcal{N}\left(\begin{pmatrix} M \\ H \end{pmatrix}\right) \quad j \geq p_l, \forall l \tag{10.6}$$

and (10.1) *is regular.*

Proof: According to Theorem 9.1, $Hx(t)$ and $Mx(t)$ have finite variance if and only if

$$\left[\bar{E}^j \left(\lambda E + F\right)^{-1} J_l\right] \Big/_{\mathcal{V}\left(\bar{E}^n\right)} \mathcal{N}\left(\bar{E}^n\right) \in \mathcal{N}\left(M\right) \quad j \geq p_l, \forall l \tag{10.7}$$

and

$$\left[\bar{E}^j \left(\lambda E + F\right)^{-1} J_l\right] \Big/_{\mathcal{V}\left(\bar{E}^n\right)} \mathcal{N}\left(\bar{E}^n\right) \in \mathcal{N}\left(H\right) \quad j \geq p_l, \forall l. \tag{10.8}$$

This gives (10.6).

□

Now, consider the problem to estimate $\bar{x}(t)$ using the Kalman filter. First note that since the disturbances $w_l(t)$ have rational spectra, they can be written as outputs from linear filters driven by white noise $v(t)$,

$$\dot{x}_w(t) = A_w x_w(t) + B_w v(t) \tag{10.9a}$$
$$w(t) = C_w x_w(t) + D_w v(t) \tag{10.9b}$$

where

$$w(t) = \begin{pmatrix} w_1(t) \\ \vdots \\ w_{n_w}(t) \end{pmatrix} \tag{10.10}$$

and $v(t)$ is white noise with variance $R_1 \delta(t)$. This should be interpreted as an SDE, see Section 2.7.2. With

$$J = \begin{pmatrix} J_1 & \cdots & J_{n_w} \end{pmatrix}, \tag{10.11}$$

(10.1) and (10.9) can be combined to give

$$\begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{x}_w(t) \end{pmatrix} = \begin{pmatrix} F & JC_w \\ 0 & A_w \end{pmatrix} \begin{pmatrix} x(t) \\ x_w(t) \end{pmatrix} + \begin{pmatrix} G \\ 0 \end{pmatrix} u(t) + \begin{pmatrix} JD_w \\ B_w \end{pmatrix} v(t) \tag{10.12a}$$

$$\bar{x}(t) = Mx(t) = \begin{pmatrix} M & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ x_w(t) \end{pmatrix} \tag{10.12b}$$

$$y(t_k) = Hx(t_k) + e(t_k) = \begin{pmatrix} H & 0 \end{pmatrix} \begin{pmatrix} x(t_k) \\ x_w(t_k) \end{pmatrix} + e(t_k). \tag{10.12c}$$

Assuming that the SDAE is regular, Theorem 2.3 can be used to transform this description into the form

$$\begin{pmatrix} x(t) \\ x_w(t) \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{10.13a}$$

$$\dot{x}_1(t) = Ax_1(t) + G_1u(t) + J_1v(t) \tag{10.13b}$$

$$\begin{pmatrix} M & 0 \end{pmatrix} Q_2x_2(t) = -\begin{pmatrix} M & 0 \end{pmatrix} Q_2 \left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}} N^{m-1} \right) G_2u(t) \tag{10.13c}$$

$$\begin{pmatrix} H & 0 \end{pmatrix} Q_2x_2(t) = -\begin{pmatrix} H & 0 \end{pmatrix} Q_2 \left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}} N^{m-1} \right) G_2u(t) \tag{10.13d}$$

provided that the estimation problem is well-posed so that $Hx(t)$ and $Mx(t)$ do not contain white noise components. Together with the measurement equation

$$y(t_k) = Hx(t_k) + e(t_k) = \begin{pmatrix} H & 0 \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} x_1(t_k) \\ x_2(t_k) \end{pmatrix} + e(t_k) \tag{10.14}$$

this finally gives the state-space description

$$\dot{x}_1(t) = Ax_1(t) + G_1u(t) + J_1v(t) \tag{10.15a}$$

$$y(t_k) = \begin{pmatrix} H & 0 \end{pmatrix} Q_1x_1(t_k) - \begin{pmatrix} H & 0 \end{pmatrix} Q_2 \left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}} N^{m-1} \right) G_2u(t_k) + e(t_k). \tag{10.15b}$$

This state-space description gives a filtering problem with continuous-time dynamics and discrete-time measurements. The Kalman filter for this setting provided, e.g., by Jazwinski (1970), can be used to estimate x_1. The estimate of $\bar{x}$ is then computed from the estimate of x_1 and the deterministic input using

$$\bar{x}(t) = \begin{pmatrix} M & 0 \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{10.16}$$

and (10.13c).

10.3 Formulations with Continuous-Time White Noise

For stochastic state-space systems, the case with a white noise input and continuous-time measurements is often considered. We will therefore consider this problem also for DAE

systems. We will thus examine the SDAE

$$E\dot{x}(t) = Fx(t) + Gu(t) + Jv(t) \tag{10.17a}$$
$$y(t) = Hx(t) + e(t) \tag{10.17b}$$
$$x(t_0) = x_0 \tag{10.17c}$$
$$\dim x(t) = n \tag{10.17d}$$

where the stochastic processes v and e are continuous-time white noise. It is assumed that the system is regular. Also here we collect the (linear combination) of variables that are of interest in a vector $\bar{x}$,

$$\bar{x} = Mx. \tag{10.18}$$

To be able to estimate the variables $\bar{x}$, we must as before require that they have finite variance. However, continuous-time Kalman filtering theory allows that the output y contains white noise signals, but not any derivatives of white noise (which would not be well-defined). In this case we therefore define well-posedness as follows.

Definition 10.2 (Well-posedness). Let $x(t)$ be defined as the solution to (10.17). The problem to estimate $\bar{x}(t) = Mx(t)$ from $y(t)$ and $u(t)$, $t \in [t_0, T]$, is *well-posed* if $Mx(t)$ has finite variance, $Hx(t)$ does not contain derivatives of white noise, and (10.17) is regular.

We shall find that a well-posed estimation problem with white noise inputs can be solved using a Kalman filter. As discussed previously, the initial value x_0 may not be chosen freely. The possibly conflicting values in x_0 will be ignored, and actually have no consequence for the computation of $x(t)$ for $t > t_0$.

Well-posedness is characterized by the following theorem.

Theorem 10.2
Consider (10.17). *Let λ be a scalar such that $(\lambda E + F)$ is invertible. Let*

$$\bar{E} = (\lambda E + F)^{-1} E. \tag{10.19}$$

Then the estimation problem (10.17) *is well-posed if and only if*

$$\left[\bar{E}^j (\lambda E + F)^{-1} J\right] \big/_{\mathcal{V}(\bar{E}^n)} \mathcal{N}(\bar{E}^n) \in \mathcal{N}(M), \quad j \geq 0 \tag{10.20a}$$
$$\left[\bar{E}^j (\lambda E + F)^{-1} J\right] \big/_{\mathcal{V}(\bar{E}^n)} \mathcal{N}(\bar{E}^n) \in \mathcal{N}(H), \quad j \geq 1. \tag{10.20b}$$

are satisfied and (10.17) *is regular.*

Proof: (10.20a) follows directly from Theorem 9.1 since white noise has pole excess $p_l = 0$. To derive (10.20b), we examine (B.8) in the proof of Theorem 9.1 in Appendix B,

$$x_a(t) = -\left(I + \left(\frac{d}{dt} + \lambda\right) N + \cdots + \left(\frac{d}{dt} + \lambda\right)^{m-1} N^{m-1}\right)\left(G_a u(t) + J_a w(t)\right). \tag{10.21}$$

Note that all J-matrices can be grouped together since all noise signals have the same pole excess. Since

$$Hx(t) = HT_1 x_s(t) + HT_2 x_a(t) \tag{10.22}$$

(with notation from Appendix B) it must be required that

$$HT_2N^jJ_a = 0, \quad j \geq 1 \tag{10.23}$$

to avoid derivatives of white noise. Now, (10.23) can be rewritten as

$$\begin{aligned}
0 &= HT_2N^jJ_a \\
&= H\begin{pmatrix}0 & T_2\end{pmatrix}\begin{pmatrix}T_1 & T_2\end{pmatrix}^{-1}\left[T_1E_s^jJ_s + T_2N^jJ_a\right] \\
&= H\left[T_1E_s^jJ_s + T_2N^jJ_a\right]\big/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2) \\
&= H\left[\begin{pmatrix}T_1 & T_2\end{pmatrix}\begin{pmatrix}E_s^j & 0\\ 0 & N^j\end{pmatrix}\begin{pmatrix}J_s\\ J_a\end{pmatrix}\right]\Big/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2) \\
&= H\left[\bar{E}^j(\lambda E + F)^{-1}J\right]\big/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2)
\end{aligned} \tag{10.24}$$

which gives (10.20b) since $\mathcal{V}(T_2(\theta)) = \mathcal{N}(\bar{E}^n(\theta))$ and $\mathcal{V}(T_1(\theta)) = \mathcal{V}(\bar{E}^n(\theta))$.

□

To see how a Kalman filter can be formulated, we rewrite (10.17) using Theorem 2.3. Under the assumption of well-posedness, this takes the form

$$x(t) = \begin{pmatrix}Q_1 & Q_2\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} \tag{10.25a}$$

$$\dot{x}_1(t) = Ax_1(t) + G_1u(t) + J_1v(t) \tag{10.25b}$$

$$MQ_2x_2(t) = -MQ_2\left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}}N^{m-1}\right)G_2u(t) \tag{10.25c}$$

$$HQ_2x_2(t) = -HQ_2\left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}}N^{m-1}\right)G_2u(t) - HQ_2J_2v(t) \tag{10.25d}$$

$$y(t) = Hx(t) + e(t) = H\begin{pmatrix}Q_1 & Q_2\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} + e(t). \tag{10.25e}$$

Inserting (10.25d) into (10.25e) gives the state-space description

$$\dot{x}_1(t) = Ax_1(t) + G_1u(t) + J_1v(t) \tag{10.26a}$$

$$y(t) = HQ_1x_1(t) - HQ_2\left(I + \cdots + \frac{d^{m-1}}{dt^{m-1}}N^{m-1}\right)G_2u(t) - HQ_2J_2v(t) + e(t). \tag{10.26b}$$

This state-space description gives a continuous-time filtering problem with correlated process and measurement noise. The Kalman filter for this problem which is given by, e.g., Kailath et al. (2000), can be used to estimate x_1. The estimate of $\bar{x}$ is then computed from the estimate of x_1 and the deterministic input using

$$\bar{x}(t) = \begin{pmatrix}M & 0\end{pmatrix}\begin{pmatrix}Q_1 & Q_2\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} \tag{10.27}$$

and (10.25c).

10.4 Example

This section presents an example that demonstrates the principles of the results discussed in the chapter. Consider two bodies, each with unit mass, moving in one dimension with velocities v_1 and v_2 and subject to external forces w_1 and w_2 respectively. If the two bodies are joined together the situation is described by the following set of equations

$$\begin{aligned} \dot{v}_1(t) &= f(t) + w_1(t) \\ \dot{v}_2(t) &= -f(t) + w_2(t) \\ 0 &= v_1(t) - v_2(t) \end{aligned} \tag{10.28}$$

where f is the force acting between the bodies. It is typical of the models obtained when joining components from model libraries that too many variables are included. (In this simple case it is of course obvious to the human modeler that this model can be simplified to that of a body with mass 2 accelerated by $w_1 + w_2$.) In the notation of (10.1) we have, with

$$x = \begin{pmatrix} v_1 \\ v_2 \\ f \end{pmatrix},$$

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad F = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix} \quad G = 0 \quad J_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad J_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

With $\lambda = 1$ we get

$$\bar{E} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

which gives

$$\mathcal{R}(\bar{E}^3) = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right\} \quad \mathcal{N}(\bar{E}^3) = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

Using the condition (10.6) we get that

$$\left[\bar{E}^j(\lambda E + F)^{-1} J_1\right] /_{\mathcal{R}(\bar{E}^3)} \mathcal{N}(\bar{E}^3) = \begin{cases} \frac{1}{2}\left(\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right) & j = 0 \\ 0 & j > 0. \end{cases}$$

$$\left[\bar{E}^j(\lambda E + F)^{-1} J_2\right] /_{\mathcal{R}(\bar{E}^3)} \mathcal{N}(\bar{E}^3) = \begin{cases} \frac{1}{2}\left(\begin{smallmatrix} 0 \\ 0 \\ -1 \end{smallmatrix}\right) & j = 0 \\ 0 & j > 0. \end{cases}$$

If w_1 and w_2 are white noise, the conditions of 10.2 are satisfied as soon as the last column of M is zero, showing that all linear combinations of v_1 and v_2 are well-defined with finite variance. If both w_1 and w_2 have pole excess greater than zero, all H and M satisfy the conditions of Theorem 10.1.

10.5 Time-Varying Linear SDAE Models

In this section we will study well-posedness of state estimation problems for time-varying linear SDAE systems. First consider a system with a white noise input $v(t)$ and white measurement noise $e(t)$,

$$E(t)\dot{x}(t) = F(t)x(t) + G(t)u(t) + J(t)v(t) \tag{10.29a}$$
$$y(t) = H(t)x(t) + e(t) \tag{10.29b}$$
$$\dim x(t) = n. \tag{10.29c}$$

We will examine when it is possible to compute an estimate of a linear combination $\bar{x}(t) = M(t)x(t)$ of the internal variables. To do this it is useful to examine the system transformed into the form described by Theorem 2.6,

$$\dot{x}_1(t) = A_{13}(t,\theta)x_3(t) + f_1(t,\theta) \tag{10.30a}$$
$$0 = x_2(t) + f_2(t,\theta) \tag{10.30b}$$
$$0 = f_3(t,\theta) \tag{10.30c}$$

where

$$\begin{pmatrix} f_1(t,\theta) \\ f_2(t,\theta) \\ f_3(t,\theta) \end{pmatrix} = P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})\big(G(t)u(t) + J(t)v(t)\big). \tag{10.31}$$

We will assume that the system is regular, so $x_3(t)$ and $f_3(t)$ are of size zero. Theorem 2.6 gives that $\dot{x}_1(t)$ is not affected by derivatives of $u(t)$ and $v(t)$ so it has finite variance and is defined by the SDE

$$\dot{x}_1(t) = f_1(t,\theta). \tag{10.32}$$

We also have that

$$y(t) = H(t)x(t) + e(t) = H(t)Q(t)\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + e(t). \tag{10.33}$$

(10.32) and (10.33) can be used to compute a filter estimate of $x_1(t)$ using Kalman filtering techniques provided that $y(t)$ does not depend on derivatives of $v(t)$. To avoid this, we must make sure that

$$H(t)Q(t)\begin{pmatrix} 0 \\ -f_2(t) \end{pmatrix} = \\ H(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})\big(G(t)u(t) + J(t)v(t)\big) \tag{10.34}$$

does not differentiate $v(t)$, or equivalently that

$$H(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J(t) \tag{10.35}$$

is of order zero in $\frac{d}{dt}$. Note that the derivatives should be handled as operators when applied to J, for example

$$\frac{d}{dt}t = 1 + t\frac{d}{dt}. \tag{10.36}$$

An estimate of $\bar{x}(t)$ can then be computed from

$$\bar{x}(t) = M(t)x(t) = M(t)Q(t)\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = M(t)Q(t)\begin{pmatrix} x_1(t) \\ 0 \end{pmatrix} + \\ M(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})\big(G(t)u(t) + J(t)v(t)\big) \tag{10.37}$$

if no white noise terms $v(t)$ occur in the expression so it only is a function of the estimated $x_1(t)$ and the known input $u(t)$. To avoid white noise terms from occurring, it must be required that

$$M(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J(t) \tag{10.38}$$

is zero, where the derivatives as before should be handled as operators. This discussion leads to the following proposition.

Proposition 10.1
Consider the regular time-varying linear DAE model (10.29) *with* $v(t)$ *and* $e(t)$ *considered as white noises. A filter estimate of* $\bar{x}(t) = M(t)x(t)$ *can be computed using standard Kalman filtering techniques provided that*

$$H(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J(t) \tag{10.39}$$

is of order zero in $\frac{d}{dt}$ *and*

$$M(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J(t) \tag{10.40}$$

is zero, where the derivatives should be handled as operators.

As in the time-invariant case, we will also study the, perhaps more realistic, case with colored noise and sampled measurements. In this case the DAE can be written as

$$E(t)\dot{x}(t) = F(t)x(t) + G(t)u(t) + \sum_{l=1}^{n_w} J_l(t)w_l(t) \tag{10.41a}$$

$$x(t_0) = x_0 \tag{10.41b}$$

$$\dim x(t) = n \tag{10.41c}$$

where $w_l(t)$ is a Gaussian second order stationary process with spectrum $\phi_{w_l}(\omega)$ which is rational in ω with pole excess $2p_l$. An output vector is measured at sampling instants t_k,

$$y(t_k) = H(t_k)x(t_k) + e(t_k), \quad k = 1,\ldots,N \tag{10.41d}$$

where $e(t_k)$ is a Gaussian random vector. To be able to compute an estimate of $\bar{x}(t) = M(t)x(t)$ in this case, it must be guaranteed that w_l is differentiated at most $p_l - 1$ times since it has pole excess $2p_l$. This holds for both x and y since none of them can include time-continuous white noise since we have discrete-time measurements. With similar calculations as was made above, we get the following proposition.

Proposition 10.2
Consider the regular linear time-varying DAE model (10.41). *A filter estimate of* $\bar{x}(t) = M(t)x(t)$ *can be computed using standard Kalman filtering techniques provided that*

$$H(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J_l(t)\Big|_{t=t_k} \tag{10.42}$$

and

$$M(t)Q(t)\begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} P_{m+1}(t,\theta)\tilde{P}_m(t,\theta,\tfrac{d}{dt})\cdots\tilde{P}_1(t,\theta,\tfrac{d}{dt})J_l(t) \tag{10.43}$$

are at most of order $p_l - 1$, $l = 1, \ldots, n_w$ *in* $\frac{d}{dt}$, *where the derivatives should be handled as operators. The derivative should be applied before inserting* $t = t_k$.

10.6 Conclusions

We have discussed well-posedness of state estimation problems for linear SDAE systems. The main results are Theorem 10.1 and 10.2 where the cases without and with continuous-time white noise were treated. The discussion also included methods to solve the state estimation problem using the Kalman filter. We have also discussed well-posedness of state estimation problems for time-varying linear SDAEs.

11

Implementation Issues

In this chapter we discuss how the canonical forms for linear DAEs can be computed using numerical software.

11.1 Introduction

The transformations presented in Section 2.3 have been used extensively in the thesis. Their existence were proven in Section 2.3, but it was not discussed how they could actually be computed. To be able use the transformations in a numerical implementation of an identification or estimation algorithm, it is of course crucial to be able to compute them numerically in a reliable manner. We will here discuss how this computation can be performed.

The discussion will include pointers to implementations of some algorithms in the linear algebra package LAPACK (Anderson et al., 1999). LAPACK is a is a free collection of routines written in Fortran77 that can be used for systems of linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. LAPACK is more or less the standard way to solve this kind of problems, and is used in commercial software like MATLAB. For operations that can be easily implemented in for example MATLAB or *Mathematica*, such as matrix multiplication and inversion, no pointers to special implementations will be made.

Some ideas related to the method presented in this section for computing the canonical forms, have earlier been published by Varga (1992). However, the presentation here is more detailed, and is closely connected to the derivation of the canonical forms presented in Section 2.3. Furthermore, we will use software from LAPACK.

In Section 11.2 we will discuss generalized eigenvalue problems and some tools which are used for solving these problems, as these are the tools which we will use to compute the canonical forms. In Section 11.3 we then discuss how the actual computation is per-

formed. The chapter is concluded with an example, a summary of the algorithm for computing the canonical forms and a note on how the results can be used in the discrete-time case.

11.2 Generalized Eigenvalues

The computation of the canonical forms will be performed with tools that normally are used for computation of generalized eigenvalues. Therefore, some theory for generalized eigenvalues will be presented in this section. The theory presented here can be found in for example the books by Bai et al. (2000) and Golub and van Loan (1996, Section 7.7).

Consider a *matrix pencil*

$$\lambda E - J \tag{11.1}$$

where the matrices E and J are $n \times n$ with constant real elements and λ is a scalar variable. We will assume that the pencil is *regular*, that is

$$\det(\lambda E - J) \not\equiv 0 \tag{11.2}$$

with respect to λ. The *generalized eigenvalues* are defined as those λ for which

$$\det(\lambda E - J) = 0. \tag{11.3}$$

If the degree p of the polynomial $\det(\lambda E - J)$ is less than n, the pencil also has $n - p$ infinite generalized eigenvalues. This happens when $\operatorname{rank} E < n$ (Golub and van Loan, 1996, Section 7.7). We illustrate the concepts with an example.

Example 11.1: Generalized eigenvalues

Consider the matrix pencil

$$\lambda \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}. \tag{11.4}$$

We have that

$$\det\left(\lambda \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}\right) = 1 + \lambda \tag{11.5}$$

so the matrix pencil has two generalized eigenvalues, ∞ and -1.

Generalized eigenvectors will not be discussed here, the interested reader is instead referred to for example Bai et al. (2000).

Since it may be difficult to solve (11.3) for the generalized eigenvalues, different transformations of the matrices that simplifies computation of the generalized eigenvalues exist. The transformations are of the form

$$P(\lambda E - J)Q \tag{11.6}$$

with invertible matrices P and Q. Such transformations do not change the eigenvalues since

$$\det\big(P(\lambda E - J)Q\big) = \det(P)\det(\lambda E - J)\det(Q). \tag{11.7}$$

One such form is the *Kronecker canonical form* of a matrix pencil discussed by e.g., Gantmacher (1960) and Kailath (1980). However, this form cannot in general be computed numerically in a reliable manner (Bai et al., 2000). For example it may change discontinuously with the elements of the matrices E and J. The transformation which we will use here is therefore instead the *generalized Schur form* which requires fewer operations and is more stable to compute (Bai et al., 2000).

The generalized Schur form of a real matrix pencil is a transformation

$$P(\lambda E - J)Q \tag{11.8}$$

where PEQ is upper quasi-triangular, that is it is upper triangular with some 2 by 2 blocks corresponding to complex generalized eigenvalues on the diagonal and PJQ is upper triangular. P and Q are orthogonal matrices. The generalized Schur form can be computed with the LAPACK commands `dgges` or `sgges`. These commands also give the possibility to sort certain generalized eigenvalues to the lower right. An algorithm for ordering of the generalized eigenvalues is also discussed by Sima (1996). Here we will use the possibility to sort the infinite generalized eigenvalues to the lower right.

The generalized Schur form discussed here is also called the generalized *real* Schur form, since the original and transformed matrices only contain real elements.

11.3 Computation of the Canonical Forms

The discussion in this section is based on the steps of the proof of the form in Theorem 2.3. We therefore begin by examining how the diagonalization in Lemma 2.1 can be performed numerically.

The goal is to find matrices P_1 and Q_1 such that

$$P_1(\lambda E - J)Q_1 = \lambda \begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} - \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix} \tag{11.9}$$

where E_1 is non-singular, E_3 is upper triangular with all diagonal elements zero and J_3 is non-singular and upper triangular. This is exactly the form we get if we compute the generalized Schur form with the infinite generalized eigenvalues sorted to the lower right. This computation can be performed with the LAPACK commands `dgges` or `sgges`. In version 7 and higher of MATLAB, the functions `qz` and `ordqz` can be used. E_1 corresponds to finite generalized eigenvalues and is therefore non-singular and E_3 corresponds to infinite generalized eigenvalues and is upper triangular with zero diagonal elements. J_3 is non-singular (and thus upper triangular with non-zero diagonal elements), otherwise the pencil would not be regular.

The next step is to compute the matrices L and R in Lemma 2.2, that is we want to solve the system

$$\begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} = \begin{pmatrix} E_1 & 0 \\ 0 & E_3 \end{pmatrix} \tag{11.10a}$$

$$\begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} = \begin{pmatrix} J_1 & 0 \\ 0 & J_3 \end{pmatrix}. \tag{11.10b}$$

Performing the matrix multiplication on the left hand side of the equations yields

$$\begin{pmatrix} E_1 & E_1R + E_2 + LE_3 \\ 0 & E_3 \end{pmatrix} = \begin{pmatrix} E_1 & 0 \\ 0 & E_3 \end{pmatrix} \tag{11.11a}$$

$$\begin{pmatrix} J_1 & J_1R + J_2 + LJ_3 \\ 0 & J_3 \end{pmatrix} = \begin{pmatrix} J_1 & 0 \\ 0 & J_3 \end{pmatrix} \tag{11.11b}$$

which is equivalent to the system

$$E_1R + LE_3 = -E_2 \tag{11.12a}$$

$$J_1R + LJ_3 = -J_2. \tag{11.12b}$$

Equation (11.12) is a *generalized Sylvester equation* (Kågström, 1994). The generalized Sylvester equation (11.12) can be solved from the linear system of equations (Kågström, 1994)

$$\begin{pmatrix} I_n \otimes E_1 & E_3^T \otimes I_m \\ I_n \otimes J_1 & J_3^T \otimes I_m \end{pmatrix} \begin{pmatrix} \operatorname{stack}(R) \\ \operatorname{stack}(L) \end{pmatrix} = \begin{pmatrix} -\operatorname{stack}(E_2) \\ -\operatorname{stack}(J_2) \end{pmatrix}. \tag{11.13}$$

Here I_n is an identity matrix with the same size as E_3 and J_3, I_m is an identity matrix with the same size as E_1 and J_1, $\otimes$ represents the Kronecker product and $\operatorname{stack}(X)$ denotes an ordered stack of the columns of a matrix X from left to right starting with the first column.

The system (11.13) can be quite large, so it may be a better choice to solve the generalized Sylvester equation (11.12) using specialized software such as the LAPACK routines `stgsyl` or `dtgsyl`.

The steps in the proof of Lemma 2.3 and Theorem 2.3 only contain standard matrix manipulations, such as multiplication and inversion. They are straightforward to implement, and will not be discussed further here.

11.4 Summary

In this section a summary of the steps to compute the canonical forms is provided. It can be used to implement the computations without studying Section 11.3 in detail. The summary is provided as a numbered list with the necessary computations.

1. Start with a system

$$E\dot{x}(t) = Jx(t) + Ku(t) \tag{11.14a}$$

$$y(t) = Lx(t) \tag{11.14b}$$

that should be transformed into

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\dot{x}(t) = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} Q^{-1}x(t) + \begin{pmatrix} B \\ D \end{pmatrix} u(t) \tag{11.15}$$

or

$$\dot{x}_1(t) = Ax_1(t) + Bu(t) \tag{11.16a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{11.16b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{11.16c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{11.16d}$$

2. Compute the generalized Schur form of the matrix pencil $\lambda E - J$ so that

$$P_1(\lambda E - J)Q_1 = \lambda \begin{pmatrix} E_1 & E_2 \\ 0 & E_3 \end{pmatrix} - \begin{pmatrix} J_1 & J_2 \\ 0 & J_3 \end{pmatrix}. \tag{11.17}$$

The generalized eigenvalues should be sorted so that diagonal elements of E_1 are non-zero and the diagonal elements of E_3 are zero. This computation can be made with one of the LAPACK commands `dgges` and `sgges`. In version 7 and higher of MATLAB, the functions `qz` and `ordqz` can be used.

3. Solve the generalized Sylvester equation

$$E_1 R + LE_3 = -E_2 \tag{11.18a}$$

$$J_1 R + LJ_3 = -J_2. \tag{11.18b}$$

to get the matrices L and R. The generalized Sylvester equation (11.18) can be solved from the linear equation system

$$\begin{pmatrix} I_n \otimes E_1 & E_3^T \otimes I_m \\ I_n \otimes J_1 & J_3^T \otimes I_m \end{pmatrix} \begin{pmatrix} \mathrm{stack}(R) \\ \mathrm{stack}(L) \end{pmatrix} = \begin{pmatrix} -\mathrm{stack}(E_2) \\ -\mathrm{stack}(J_2) \end{pmatrix} \tag{11.19}$$

or with the LAPACK commands `stgsyl` or `dtgsyl`. Here I_n is an identity matrix with the same size as E_3 and J_3, I_m is an identity matrix with the same size as E_1 and J_1, $\otimes$ represents the Kronecker product and $\mathrm{stack}(X)$ denotes an ordered stack of the columns of a matrix X from left to right starting with the first column.

4. We now get the form (11.15) and (11.16) according to

$$P = \begin{pmatrix} E_1^{-1} & 0 \\ 0 & J_3^{-1} \end{pmatrix} \begin{pmatrix} I & L \\ 0 & I \end{pmatrix} P_1 \tag{11.20a}$$

$$Q = Q_1 \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \tag{11.20b}$$

$$N = J_3^{-1} E_3 \tag{11.20c}$$

$$A = E_1^{-1} J_1 \tag{11.20d}$$

$$\begin{pmatrix} B \\ D \end{pmatrix} = PK. \tag{11.20e}$$

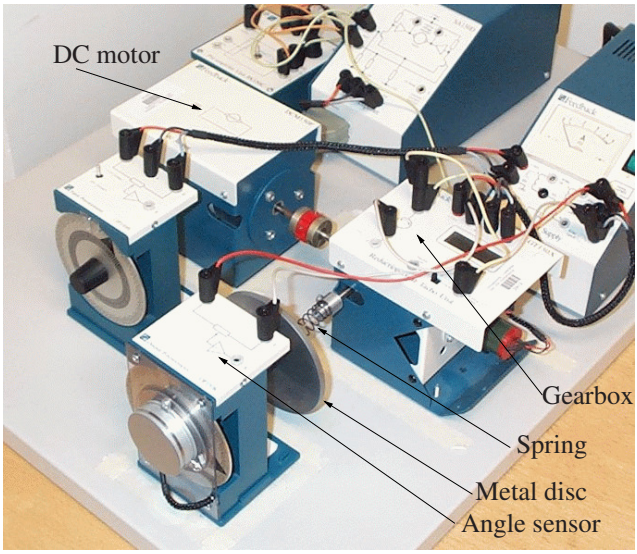


Figure 11.1: *The examined process.*

11.5 Application Example

In this section it is exemplified how the algorithms presented in this chapter can be used when implementing parameter estimation for a physical process. The system setup examined is a DC-motor connected to a heavy metal disc through a gearbox and a spring, see Figure 11.1. This setup simulates the problems that occur in power transfer in weak axes, such as the rear axis in trucks. This problem is studied within the area of power train control.

It should be stressed that the role of this example is to show how we can work with DAE descriptions from Modelica-like modeling environments in estimation applications. In this case, despite a singular E-matrix, the model will be reduced to a standard state-space description by the transformation mechanisms described in the earlier sections. The properties of the actual estimates obtained will thus follow from well-known techniques and results, and we will therefore not discuss accuracy aspects of the estimated models.

The laboratory process was modeled in Modelica. The model is linear, so the resulting equations can be written on the form

$$E(\theta)\dot{x}(t) = F(\theta)x(t) + G(\theta)u(t) \tag{11.21a}$$

$$y(t) = H(\theta)x(t). \tag{11.21b}$$

The actual estimation was performed using MATLAB. The transformation of the Modelica model into DAE form in MATLAB was performed manually, but the procedure could quite easily be automated if it were possible to specify inputs, outputs, and unknown parameters in Modelica. This is an important subject for future work, since gray-box identification then could be performed by first modeling the system using Modelica, and then estimating

unknown parameters and states in MATLAB without having to manipulate any equations manually.

To use, e.g., the System Identification Toolbox for MATLAB (Ljung, 2006) to estimate unknown parameters, the model was put into the `idgrey` object format. This means that an m-file must be written, which, for each parameter vector produces the matrices of a linear state-space system, $A, B, C, D, K, x0$. This means that this m-file will call the transformation routines described previously in the chapter, which include calls to functions from LAPACK. The model object is created by

```
mi = idgrey('servo',[10 -10],'c',[], ...
     0,'DisturbanceModel','None');
```

and the model-defining m-file `servo` has the format

```
function [A,B,C,D,K,X0]=servo(pars,Tsm,Auxarg)

%Get DAE matrices E,F,G & H with
%parameters above
[E,F,G,H]=SpringServoMatrices(pars);

% Call to Lapack routine
[A,B,C,D]=StandardForm(E,F,G,H);

K=0; %Output error model
X0=0;
```

The function `SpringServoMatrices` computes DAE matrices corresponding to the model structure (11.21) for a certain parameter value, and `StandardForm` computes a corresponding state-space description using the methods discussed in this chapter. In this case a well-defined state-space model is generated for all parameter values, so the estimation command

```
m = pem(data,mi)
```

will work in a familiar fashion.

11.6 Difference-Algebraic Equations

The method for computing the canonical forms for difference-algebraic equations, or discrete-time descriptor systems, is identical to the computation for continuous-time systems. This can be realized since the proofs of the transformations in the continuous-time and discrete-time cases (Chapter 2) are similar and the computation for the continuous-time case is based on the proof of the transformation. For the summary in Section 11.4, the only thing that changes is actually the first step. For the discrete-time case it takes the following form:

1. Start with a system

$$Ex(t+1) = Jx(t) + Ku(t) \tag{11.22a}$$

$$y(t) = Lx(t) \tag{11.22b}$$

that should be transformed into

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}x(t+1) = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} Q^{-1}x(t) + \begin{pmatrix} B \\ D \end{pmatrix} u(t) \tag{11.23}$$

or

$$x_1(t+1) = Ax_1(t) + Bu(t) \tag{11.24a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i Du(t+i) \tag{11.24b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{11.24c}$$

$$y(t) = LQ \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{11.24d}$$

The steps 2–4 are identical to Section 11.4.

11.7 Conclusions

We examined how the canonical forms discussed in Section 2.3 can be computed with numerical software. The calculation is based on tools for the solution of generalized eigenvalue problems, so generalized eigenvalue problems where briefly discussed. Implementations of the tools for generalized eigenvalue problems are available in the free LAPACK package.

12

Initialization of Parameter Estimates

Since DAE systems can be formed by simply writing down basic physical relations, the matrix elements of linear DAE systems are often physical parameters or known constants. This special structure is not used by the parameter estimation methods discussed in the earlier chapters. In this chapter we will discuss how to utilize the structure that models in DAE form often have to initialize parameter estimation methods.

12.1 Introduction

The parameter estimation methods discussed earlier have in common that they construct a criterion function $V(\theta)$ that should be minimized to estimate the unknown parameters. For the physically parameterized model structures discussed in this thesis, $V(\theta)$ is a complex function of the parameters θ. This means that the criterion function in general cannot be minimized analytically. Instead, we have to resort to numerical search methods such as Gauss-Newton as discussed by Ljung (1999). Such methods only guarantee convergence to a local minimum, and experience shows that it can be difficult to find the global minimum of $V(\theta)$ for physically parameterized model structures. One remedy is to use physical insight when selecting initial values for the numerical search, and another is to do several searches with different starting values. Although these remedies can work well in many cases, there is still no guarantee that the global optimum is found. In this chapter we will therefore discuss how initial parameter values for the numerical search can be chosen by minimization of a polynomial. The following example illustrates this.

Example 12.1: Initalization through transfer function coefficients.

Consider a body with mass m to which a force $F(t)$ is applied. The motion of the body is damped by friction with damping coefficient k. If $x(t)$ is the position of the body, the equation for the motion of the body is $m\ddot{x}(t) = F(t) - k\dot{x}(t)$. The position is the measured output of the model. With $v(t)$ denoting the velocity of the body, this can be

written in DAE form as

$$\begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -k \end{pmatrix} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} F(t) \tag{12.1a}$$

$$y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}. \tag{12.1b}$$

The transfer function for this system is

$$G(s,\theta) = \begin{pmatrix} 1 & 0 \end{pmatrix} \left(s \begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 0 & -k \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{ms^2 + ks}. \tag{12.2}$$

If a black-box estimation procedure has given the transfer function

$$\hat{G}(s) = \frac{1}{2s^2 + 3s + 0.01} \tag{12.3}$$

a polynomial which measures the difference of the transfer function coefficients is

$$p(\theta) = (m-2)^2 + (k-3)^2 + 0.01^2. \tag{12.4}$$

This polynomial is minimized by $m = 2$ and $k = 3$.

As shown in the example, we assume that a black-box model of the system has been estimated beforehand by for example a subspace method (Ljung, 1999). The polynomial is then formed as a measure of the "distance" between the black-box model and the physically parameterized model. Although the measure is formed in an ad hoc manner, it should in many cases give a better initial value than a pure guess. However, we will have no guarantees for the quality of the initial values selected. Therefore the results should be compared to the results for initial values selected from physical insight, or for randomly selected initial values.

We saw in Example 12.1, that if the black-box model and the physically parameterized model both are in transfer function form, one straightforward way to get initial values for the parameter search is to try to make the coefficients of the numerator and denominator polynomials as equal as possible. Note that linear state-space and linear DAE models easily can be converted to transfer functions as discussed earlier.

Although the polynomial $p(\theta)$ in Example 12.1 was trivial to minimize, one can note that $p(\theta)$ can be a high order polynomial with as many variables as there are unknown parameters. In some cases it could be preferable to have a polynomial with a lower degree, but with a higher number of variables. For parameterized state-space systems, Parrilo and Ljung (2003) discusses a method to find a polynomial which is biquadratic in its variables (this work is based on the paper by Xie and Ljung, 2002). This method requires that the elements of the state-space matrices are unknown parameters or constants. It is also proposed that the polynomial could be minimized by sum of squares optimization. The price to get a biquadratic polynomial to minimize is that more variables than the unknown parameters must be included in the polynomial.

The requirement that the elements of the state-space matrices should be unknown physical parameters or known constants can be rather strict. Since one usually needs

to make different transformations to get a state-space description, the elements of the matrices are usually functions of the unknown physical parameters. It is much more likely that the elements of the matrices of a linear DAE system are unknown parameters or constant, since basic physical equations often are simple integrators and static relations. By applying the technique from Parrilo and Ljung (2003) to linear DAE systems, we can therefore utilize the structure that often is present in linear DAE systems. This is what this chapter is about. We will also discuss *sum of squares optimization* (Parrilo, 2000; Prajna et al., 2004), which in some cases can be used to find the global minimum of a polynomial.

That linear DAE systems often have simple structure is also motivated by Example 12.2 below.

Example 12.2: DAE model versus state-space model

Consider the system in Example 12.1:

$$\begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -k \end{pmatrix} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} F(t) \tag{12.5}$$

In DAE form, the elements of the matrices are clearly known or physical parameters. However, this is not the case if the system is written in state-space form:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{k}{m} \end{pmatrix} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} F(t) \tag{12.6}$$

12.2 Transforming the Problem

In this section, we describe how the problem of finding initial values for the parameters to be estimated can be posed as the minimization of a biquadratic polynomial. The transformation is based on that we have a consistently estimated black-box model in state-space form,

$$\dot{x}(t) = A_0 x(t) + B_0 u(t) \tag{12.7a}$$
$$y(t) = C_0 x(t), \tag{12.7b}$$

which could have been estimated using for example a subspace method (Ljung, 1999). The idea is then that there should exist a transformation between the parameterized DAE model and the black-box model for the optimal parameter values. Because of modeling errors and noise, there will typically not exist an exact transformation between the systems, and we therefore choose to minimize a norm which measures the difference between the two systems as a function of the parameters.

As the transformations are simplified considerably in the special case when $E(\theta)$ is invertible, this case is discussed separately in section 12.2.1. The general case is discussed in section 12.2.2.

12.2.1 The Case of Invertible $E(\theta)$

Consider the DAE system

$$E(\theta)\dot{x}(t) = J(\theta)x(t) + K(\theta)u(t) \tag{12.8a}$$
$$y(t) = L(\theta)x(t) \tag{12.8b}$$

and let $E(\theta)$ be invertible. Lemma 2.3 gives that a transformation

$$PE(\theta)QQ^{-1}\dot{x}(t) = PJ(\theta)QQ^{-1}x(t) + PK(\theta)u(t) \tag{12.9a}$$
$$y(t) = L(\theta)QQ^{-1}x(t) \tag{12.9b}$$

with invertible P and Q results in a state-space description,

$$\dot{z}(t) = PJ(\theta)Qz(t) + PK(\theta)u(t) \tag{12.10a}$$
$$y(t) = L(\theta)Qz(t) \tag{12.10b}$$
$$x(t) = Qz(t). \tag{12.10c}$$

It is clear that it is possible to achieve all state-space descriptions that are equivalent to (12.8) in this way by including a further similarity transformation of the state-space description in P and Q.

If we now have a consistent estimate of the system in the form (12.7), we want to find parameter values θ that make the input-output behavior of (12.7) and (12.8) as equal as possible. If it were possible to make them exactly equal, there would be matrices P and Q and parameter values $\hat{\theta}$ such that

$$PE(\hat{\theta})Q = I \tag{12.11a}$$
$$PJ(\hat{\theta})Q = A_0 \tag{12.11b}$$
$$PK(\hat{\theta}) = B_0 \tag{12.11c}$$
$$L(\hat{\theta})Q = C_0 \tag{12.11d}$$

which also can be written as

$$PE(\hat{\theta}) = Q^{-1} \tag{12.12a}$$
$$PJ(\hat{\theta}) = A_0Q^{-1} \tag{12.12b}$$
$$PG(\hat{\theta}) = B_0 \tag{12.12c}$$
$$L(\hat{\theta}) = C_0Q^{-1}. \tag{12.12d}$$

As there will always be some noise and modeling errors, we cannot expect these equations to hold exactly. Therefore we form a polynomial that measures how well these equations are satisfied:

$$\begin{aligned} p_1(\theta, P, Q^{-1}) = {} & \|PE(\theta) - Q^{-1}\|_F^2 \\ & + \|PJ(\theta) - A_0Q^{-1}\|_F^2 \\ & + \|PG(\theta) - B_0\|_F^2 \\ & + \|L(\theta) - C_0Q^{-1}\|_F^2 \end{aligned} \tag{12.13}$$

Here $\|\cdot\|_F^2$ denotes the squared Frobenius norm, i.e., the sum of all squared matrix elements. This polynomial is always biquadratic in the unknown parameters θ and the elements of the matrices P and Q^{-1}, if the elements of the DAE matrices are constants or unknown parameters. When the polynomial is formed as in Example 12.1, the polynomial is not guaranteed to be biquadratic, but could have higher degree. The method in this section consequently guarantees that the polynomial to be minimized is biquadratic to the prize of a higher number of variables. If minimization of (12.13) does not give good results, one may instead try to minimize

$$\begin{aligned} p_2(\theta, P^{-1}, Q) = &\|E(\theta)Q - P^{-1}\|_F^2 \\ &+ \|J(\theta)Q - P^{-1}A_0\|_F^2 \\ &+ \|G(\theta) - P^{-1}B_0\|_F^2 \\ &+ \|L(\theta)Q - C_0\|_F^2. \end{aligned} \tag{12.14}$$

This polynomial is biquadratic in the unknown parameters θ and the elements of the matrices P^{-1} and Q if the elements of the DAE matrices are constants or unknown parameters. It also measures how well (12.11) is satisfied.

12.2.2 The Case of Non-Invertible $E(\theta)$

In the case when $E(\theta)$ is not invertible, it is still possible to formulate a polynomial that can give good initial values for the parameter search when minimized. However, in this more complex case, it cannot in general be guaranteed that the polynomial will be biquadratic in the unknown variables. Therefore we will also discuss additional assumptions to achieve this.

As the output of DAE systems can depend on derivatives of the input, we must assume that the estimated black-box model of the system is in the form

$$\dot{x}(t) = A_0 x(t) + B_0 u(t) \tag{12.15a}$$

$$y(t) = C_0 x(t) + \sum_{k=0}^{m-1} D_{0k} u^{(k)}(t). \tag{12.15b}$$

Furthermore, we know from Lemma 2.3 that for each selection of parameter values θ there exists a transformation

$$PE(\theta)QQ^{-1}\dot{x}(t) = PJ(\theta)QQ^{-1}x(t) + PK(\theta)u(t) \tag{12.16a}$$

$$y(t) = L(\theta)QQ^{-1}x(t) \tag{12.16b}$$

that gives the system

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ D \end{pmatrix} u(t) \tag{12.17a}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{12.17b}$$

$$y(t) = L(\theta)Q \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{12.17c}$$

According to Theorem 2.3 this can be further transformed into the form

$$\dot{x}_1(t) = Ax_1(t) + Bu(t) \tag{12.18a}$$

$$x_2(t) = -Du(t) - \sum_{i=1}^{m-1} N^i D u^{(i)}(t) \tag{12.18b}$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = Q^{-1}x(t) \tag{12.18c}$$

$$y(t) = L(\theta)Q \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{12.18d}$$

We now want to find parameter values θ and transformation matrices P and Q such that the models (12.15) and (12.18) have the same input-output behavior. From (12.15)–(12.18), we see that this is the case if the following equations are satisfied.

$$PE(\theta)Q = \begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \tag{12.19a}$$

$$PJ(\theta)Q = \begin{pmatrix} A_0 & 0 \\ 0 & I \end{pmatrix} \tag{12.19b}$$

$$PK(\theta) = \begin{pmatrix} B_0 \\ D \end{pmatrix} \tag{12.19c}$$

$$L(\theta)Q = \begin{pmatrix} C_0 & C_2 \end{pmatrix} \tag{12.19d}$$

$$D_{00} = -C_2 D \tag{12.19e}$$

$$D_{0k} = -C_2 N^k D, \ k = 1 \dots m-1. \tag{12.19f}$$

$$N^m = 0 \tag{12.19g}$$

Here we introduced the matrix C_2 to simplify the notation. Equation (12.19g) guarantees that N is nilpotent. This can also be achieved by for example parameterizing N as an upper triangular matrix with zero diagonal elements, but then extra care would have to be taken to guarantee that N is nilpotent of the correct order. A polynomial that measures how well these equations are satisfied can now be formed:

$$\begin{aligned} p_3(\theta, P, Q^{-1}, N, D, C_2) = {} & \|PE(\theta) - \begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\|_F^2 \\ & + \|PJ(\theta) - \begin{pmatrix} A_0 & 0 \\ 0 & I \end{pmatrix} Q^{-1}\|_F^2 \\ & + \|PK(\theta) - \begin{pmatrix} B_0 \\ D \end{pmatrix}\|_F^2 \\ & + \|L(\theta) - \begin{pmatrix} C_0 & C_2 \end{pmatrix} Q^{-1}\|_F^2 \\ & + \|D_{00} + C_2 D\|_F^2 \\ & + \sum_{k=1}^{m-1} \|D_{0k} + C_2 N^k D\|_F^2 \\ & + \|N^m\|_F^2 \end{aligned} \tag{12.20}$$

This polynomial can unfortunately not be guaranteed to be biquadratic in its variables (the elements of θ and the unknown matrices), even if the elements of the DAE matrices are constants or unknown parameters. However, if the true system has

$$D_{0k} = 0, \;\; k = 0 \ldots m-1 \tag{12.21}$$

and the DAE model is such that

$$C_2 D = 0 \tag{12.22a}$$
$$C_2 N^k D = 0, \;\; k = 1 \ldots m-1 \tag{12.22b}$$
$$N^m = 0 \tag{12.22c}$$

then (12.20) simplifies to

$$\begin{aligned} p_4(\theta, P, Q^{-1}, N, D, C_2) = {} & \|PE(\theta) - \begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} Q^{-1}\|_F^2 \\ & + \|PJ(\theta) - \begin{pmatrix} A_0 & 0 \\ 0 & I \end{pmatrix} Q^{-1}\|_F^2 \\ & + \|PK(\theta) - \begin{pmatrix} B_0 \\ D \end{pmatrix}\|_F^2 \\ & + \|L(\theta) - \begin{pmatrix} C_0 & C_2 \end{pmatrix} Q^{-1}\|_F^2. \end{aligned} \tag{12.23}$$

This polynomial is biquadratic in its variables.

The relation (12.21) can in many cases be physically motivated, since it is common that the output of physical systems does not depend directly on the input or its derivatives. If this is the case, the DAE matrices should be parameterized so that the (12.22) holds for all or almost all parameter values. Note that it always can be tested afterwards if (12.22) is fulfilled. This is simply done by testing if $C_2 D = 0$, if $C_2 N^k D = 0$, and if $N^m = 0$.

12.3 Sum of Squares Optimization

The polynomials that are formed in this chapter could be minimized by any method that gives the global minimum. One family of methods that could be used are algebraic methods, such as Gröbner bases. Here we will discuss another method which relaxes the minimization to a sum of squares problem, as described by, e.g., Parrilo (2000) and Prajna et al. (2004). To describe this procedure, we first need to note that the problem

$$\min_{\theta} p(\theta) \tag{12.24}$$

also can be written as

$$\max \quad \lambda \tag{12.25a}$$
$$\text{subject to} \quad p(\theta) - \lambda \geq 0 \quad \text{for all } \theta. \tag{12.25b}$$

Now, since a sum of squared real polynomials $f_i(\theta, \lambda)$ always is greater than or equal to zero, a relaxation of (12.25) is

$$\max \quad \lambda \tag{12.26a}$$

$$\text{subject to} \quad p(\theta) - \lambda = \sum_i f_i^2(\theta, \lambda). \tag{12.26b}$$

As described in the references, the relaxed problem always gives a lower bound on the optimal value, and for many problems it also attains a strict lower bound. The relaxed problem can be solved using semidefinite programming as described by Prajna et al. (2004). The algorithms for finding the lower bound also often find variable values that attain this lower bound, and in this case we of course have the actual optimum.

The reason that the algorithm gives a lower bound, which is not guaranteed to be the actual optimum, is that non-negativity of a polynomial is not equivalent to the polynomial being a sum of squares. However, in the following cases non-negativity and the existence of a sum of squares decomposition are equivalent (Prajna et al., 2004):

- Univariate polynomials of any (even) degree.
- Quadratic polynomials in any number of variables.
- Quartic polynomials in two variables.

Unfortunately, the polynomials we have formed are biquadratic, so we are not guaranteed to find the minimum. If the optimal value is zero we will anyway have equivalence between non-negativity and the existence of a sum of squares decomposition since the original formulation of the polynomials, since (12.13), (12.14), (12.20), or (12.23) then are suitable sum of square decompositions for $\lambda = 0$. We will have this case if there exists a parameter value that make the input-output behavior of the DAE system and black-box model exactly equal.

12.4 Difference-Algebraic Equations

The discussion in Section 12.2 is valid also for the discrete-time case. The only difference is that we have a difference-algebraic equation

$$E(\theta)x(t+1) = J(\theta)x(t) + K(\theta)u(t) \tag{12.27a}$$

$$y(t) = L(\theta)x(t) \tag{12.27b}$$

for which we need to find initial values for parameter estimation. In the case when $E(\theta)$ is invertible, we assume that a consistently estimated black-box model

$$x(t+1) = A_0x(t) + B_0u(t) \tag{12.28a}$$

$$y(t) = C_0x(t) \tag{12.28b}$$

is available. The polynomials p_1 and p_2 in (12.13) and (12.14) can then be minimized to find initial parameter values. In the case where $E(\theta)$ is not invertible, we instead assume

that a black-box model according to

$$x(t+1) = A_0 x(t) + B_0 u(t) \tag{12.29a}$$

$$y(t) = C_0 x(t) + \sum_{k=0}^{m-1} D_{0k} u(t+k) \tag{12.29b}$$

is available. The polynomial p_3 in (12.20) can then be used to find initial values. If the assumptions (12.21) and (12.22) are fulfilled, the simpler polynomial p_4 in (12.23) can be used instead.

12.5 Conclusions

We noted that the standard system identification problem often is a minimization with many local minima. As the minimization problem normally is solved using a standard optimization method, it is important to have initial values near the optimal values of parameters that are to be estimated. We noted that a polynomial which measures the difference between the coefficients of transfer functions can be formed. If this polynomial is minimized, it should give good initial values. However, this polynomial can have a high degree, so we examined how a polynomial which is biquadratic can be formed. This polynomial also gives an initial guess for the parameters if it is minimized, but has more unknown variables. To guarantee that this polynomial is biquadratic, we used the special structure that often is present in linear DAE systems.

13

Conclusions

A large part of this thesis has been devoted to noise modeling in DAE models. For nonlinear DAE models, sufficient conditions for well-posedness of stochastic models were developed. For the linear case, a more thorough analysis could be performed to develop both necessary and sufficient conditions for well-posedness of models and different estimation problems.

The motivation to study DAE models was mainly that component-based modeling, such as in Modelica, leads to DAE models. The motivation for studying noise models, is that they allow implementation of estimation methods that have proven to be effective for estimation of time dependent variables and time invariant parameters. It was consequently also discussed how the stochastic DAE models could be used for particle filtering, Kalman filtering, and parameter estimation using the maximum likelihood and prediction error methods. It was also suggested how the methods could be implemented. For nonlinear DAE models, it was suggested to use DAE solvers. Here further work could be directed at utilizing the structure of the equations to speed up the computations. For linear models, it was discussed how the methods in the thesis can be implemented using tools from the linear algebra package LAPACK.

We have also examined model properties such as observability and identifiability, which are important in connection with parameter and state estimation. These properties were studied for nonlinear DAE models, but the results are of course valid also for linear models. A basic idea was to formulate the observability problem itself as a DAE, and examine the properties of that DAE. As we have seen, this idea can also be used for example to examine zero dynamics. An interesting topic to future research is to examine if this idea can be used for analysis of other model properties as well.

For linear models, the problem of finding initial values for parameter estimation procedures was briefly discussed. This is an important topic where more research is necessary, both for linear and nonlinear models.

Appendices

A

Notation

Symbols and Mathematical Notation

$\mathbb{R}$	The set of real numbers
t	Time variable
$\dot{x}(t)$	Derivative of the function $x(t)$ with respect to time, $\frac{d}{dt}x(t)$
$\ddot{x}(t)$	The second derivative of the function $x(t)$ with respect to time, $\frac{d^2}{dt^2}x(t)$
$x^{(n)}(t)$	The n:th derivative of the function $x(t)$ with respect to time, $\frac{d^n}{dt^n}x(t)$
$px(t)$	Derivative of the function $x(t)$ with respect to time, $\frac{d}{dt}x(t)$
$qx(t_k)$	Shift operator, $qx(t_k) = x(t_{k+1})$
$\delta(t)$	Generalized Dirac function
δ_{t_k,t_s}	$\delta_{t_k,t_s} = 1$ if $t_k = t_s$, $\delta_{t_k,t_s} = 0$ otherwise
$\arg\min_x f(x)$	The value of x that minimizes the function $f(x)$
$\arg\max_x f(x)$	The value of x that maximizes the function $f(x)$
$\operatorname{rank} A$	Rank of the matrix A
$\operatorname{corank} A$	Corank of the matrix A
$\operatorname{kernel} A$	Kernel (null space) of the matrix A
$\mathcal{N}(A)$	Kernel (null space) of the matrix A
$\operatorname{cokernel} A$	Cokernel of the matrix A
$\operatorname{range} A$	Range of the matrix A
$\mathcal{V}(A)$	Range of the matrix A
$\operatorname{corange} A$	Corange of the matrix A

$\det(\cdot)$	The determinant of the argument
$\otimes$	Kronecker product
$\mathrm{stack}(\cdot)$	An ordered stack of the columns of the (matrix) argument from left to right starting with the first column
A^*	Conjugate transpose of the matrix A
I	Identity matrix of appropriate dimensions
$\dim x$	Dimension of the vector x
$A/_B C$	Oblique projection of the matrix A along the space B on the space C
θ	Vector of unknown variables in a system identification problem
$D_{\mathcal{M}}$	Set in which the parameters θ lie
Z^N	Measured data, $\{u(t_0), y(t_0), ..., u(t_N), y(t_N)\}$ or $\{U(\omega_1), Y(\omega_1), ..., U(\omega_N), Y(\omega_N)\}$
$\hat{y}(t_k \vert t_{k-1}, \theta)$	A model's prediction of $y(t_k)$ given θ and Z^{k-1}
$\varepsilon(t_k, \theta)$	Prediction error, $y(t_k) - \hat{y}(t_k \vert t_{k-1}, \theta)$
$\mathrm{E}(x)$	Expected value of the stochastic variable x
Pr	Probability
$\mathrm{cov}(x, y)$	Covariance for the stochastic variables x and y
$\mathrm{var}(x)$	Variance of the stochastic variable x
$r_{xy}(s, t)$	Covariance function for the stochastic processes $x(s)$ and $y(t)$
$\mathcal{L}[\cdot]$	Laplace transform of the argument
$\mathcal{Z}[\cdot]$	Z transform of the argument
F_l	Derivative array
$F_{l;p}$	Partial derivatives of F_l with respect to the variables p

Acronyms

DAE	Differential-algebraic equation
MFD	Matrix fraction description
MSS	Minimally structurally singular
RMSE	Root mean square error
SDAE	Stochastic differential-algebraic equation
SDE	Stochastic differential equation
SVD	Singular value decomposition

B

Proof of Theorem 9.1

In this appendix Theorem 9.1 is proved. Recall that $\lambda(\theta)$ is a scalar such that $\lambda(\theta)E(\theta) + F(\theta)$ is invertible and

$$\bar{E}(\theta) = \big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1}E(\theta). \tag{B.1}$$

First we will prove two propositions:

Proposition B.1
Consider the SDAE (9.1) *with the matrix* $\bar{E}(\theta)$ *transformed into Jordan form:*

$$\bar{E}(\theta) = \underbrace{\begin{pmatrix} T_1(\theta) & T_2(\theta) \end{pmatrix}}_{T(\theta)} \begin{pmatrix} E_s(\theta) & 0 \\ 0 & N(\theta) \end{pmatrix} \begin{pmatrix} T_1(\theta) & T_2(\theta) \end{pmatrix}^{-1} \tag{B.2}$$

where the zero eigenvalues are sorted to the lower right so that E_s *is invertible and* N *is nilpotent of order* m *(*$N^m = 0$*,* $N^{m-1} \neq 0$*).*

Then the transformation

$$x = \underbrace{\begin{pmatrix} T_1(\theta) & T_2(\theta) \end{pmatrix}}_{T(\theta)} \begin{pmatrix} x_s \\ x_a \end{pmatrix} \tag{B.3}$$

gives a system description of the form

$$E_s(\theta)\dot{x}_s = \big(I - \lambda(\theta)E_s(\theta)\big)x_s + G_s(\theta)u + \sum_{l=1}^{n_w} J_{l,s}(\theta)w_l(\theta) \tag{B.4a}$$

$$N(\theta)\dot{x}_a = \big(I - \lambda(\theta)N(\theta)\big)x_a + G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta) \tag{B.4b}$$

where

$$\begin{pmatrix} J_{l,s}(\theta) \\ J_{l,a}(\theta) \end{pmatrix} = T^{-1}(\theta)\big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1} J_l(\theta) \tag{B.5}$$

$$\begin{pmatrix} G_s(\theta) \\ G_a(\theta) \end{pmatrix} = T^{-1}(\theta)\big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1} G(\theta). \tag{B.6}$$

Proof: Adding $\lambda(\theta)E(\theta)x$ to each side of Equation (9.1a) and then multiplying from the left with $(\lambda E(\theta) + F(\theta))^{-1}$ gives

$$\bar{E}(\theta)\big(\dot{x} + \lambda(\theta)x\big) \;=\; x + \big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1} \times \left(G(\theta)u + \sum_{l=1}^{n_w} J_l(\theta)w_l(\theta)\right).$$

Substituting

$$x = T(\theta)\begin{pmatrix} x_s \\ x_a \end{pmatrix} \tag{B.7}$$

and multiplying from the left with $T^{-1}(\theta)$ gives

$$T^{-1}(\theta)\bar{E}(\theta)T(\theta)\left(\begin{pmatrix} \dot{x}_s \\ \dot{x}_a \end{pmatrix} + \lambda\begin{pmatrix} x_s \\ x_a \end{pmatrix}\right) = \\ \begin{pmatrix} x_s \\ x_a \end{pmatrix} + T^{-1}(\theta)\big(\lambda E(\theta) + F(\theta)\big)^{-1} \times \left(G(\theta)u + \sum_{l=1}^{n_w} J_l(\theta)w_l(\theta)\right)$$

which is the desired form. □

Proposition B.2
The auxiliary variables x_a can be solved from (B.4b) *to give*

$$x_a = -\Bigg(I + \Big(\frac{d}{dt} + \lambda(\theta)\Big)N(\theta) + \cdots + \\ \Big(\frac{d}{dt} + \lambda(\theta)\Big)^{m-1} N^{m-1}(\theta)\Bigg) \times \left(G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta)\right) \tag{B.8}$$

Proof: Writing (B.4b) as

$$x_a = N(\theta)\left(\frac{d}{dt} + \lambda(\theta)\right)x_a - \left(G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta)\right) \tag{B.9}$$

and successively "multiplying" by $N(\theta)(\frac{d}{dt} + \lambda)$ gives (omitting dependence on θ)

$$N\left(\frac{d}{dt} + \lambda\right)x_a = N^2\left(\frac{d}{dt} + \lambda\right)^2 x_a - N\left(\frac{d}{dt} + \lambda\right)\left(G_a u + \sum_{l=1}^{n_w} J_{l,a}w_l(\theta)\right)$$

$$\vdots$$

$$N^{m-1}\left(\frac{d}{dt} + \lambda\right)^{m-1} x_a = -N^{m-1}\left(\frac{d}{dt} + \lambda\right)^{m-1}\left(G_a u + \sum_{l=1}^{n_w} J_{l,a}w_l\right)$$

where we have used $N^m = 0$ in the last equation. A successive substitution from these equations into (B.9) then gives (B.8). $\square$

We now prove the main result, Theorem 9.1.

Proof: Transforming the system into the form (B.4) we see that x_s is an integration of the second order processes $w_l(\theta)$. Hence, it has finite variance. Since

$$H(\theta)x = H(\theta)T_1(\theta)x_s + H(\theta)T_2(\theta)x_a$$

it must also be required that $H(\theta)T_2(\theta)x_a$ has finite variance. Note that $w_l(\theta)$ has finite variance if it is differentiated at most $p_l - 1$ times since it has pole excess $2p_l$. This can be realized from (2.179b) which gives that the variance of $\frac{d^n}{dt^n} w_l(\theta)$ is

$$r(0) = \int_{-\infty}^{\infty} (iw)^{2n} \phi_{w_l} d\omega \begin{cases} < \infty & \text{if } n \leq p_l - 1 \\ = \infty & \text{if } n \geq p_l. \end{cases} \tag{B.10}$$

(B.8) thus gives that $H(\theta)T_2(\theta)x_a$ has finite variance if and only if

$$H(\theta)T_2(\theta)N^j(\theta)J_{l,a}(\theta) = 0 \quad j \geq p_l, \forall l. \tag{B.11}$$

By using the notation $[\cdot]/_{\mathcal{X}}\,\mathcal{Y}$ for the oblique projection on the space $\mathcal{Y}$ along the space $\mathcal{X}$ and $\mathcal{V}(A)$ for the space spanned by the columns of the matrix A, this condition can be written as (omitting dependence on θ)

$$\begin{aligned} 0 &= HT_2N^jJ_{l,a} \\ &= H\begin{pmatrix} 0 & T_2 \end{pmatrix}\begin{pmatrix} T_1 & T_2 \end{pmatrix}^{-1}\left[T_1E_s^jJ_{l,s} + T_2N^jJ_{l,a}\right] \\ &= H\left[T_1E_s^jJ_{l,s} + T_2N^jJ_{l,a}\right]/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2) \\ &= H\left[\begin{pmatrix} T_1 & T_2 \end{pmatrix}\begin{pmatrix} E_s^j & 0 \\ 0 & N^j \end{pmatrix}\begin{pmatrix} J_{l,s} \\ J_{l,a} \end{pmatrix}\right]\Big/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2) \\ &= H\left[\bar{E}^j(\lambda E + F)^{-1}J_l\right]/_{\mathcal{V}(T_1)}\,\mathcal{V}(T_2). \end{aligned} \tag{B.12}$$

Since $E_s(\theta)$ is invertible and $N(\theta)$ is nilpotent, (B.2) gives that $\mathcal{V}(T_2(\theta)) = \mathcal{N}(\bar{E}^n(\theta))$ and that $\mathcal{V}(T_1(\theta)) = \mathcal{V}(\bar{E}^n(\theta))$, so the condition can also be written

$$\left[\bar{E}^j(\theta)\big(\lambda(\theta)E(\theta) + F(\theta)\big)^{-1}J_l(\theta)\right]\Big/_{\mathcal{V}\left(\bar{E}^n(\theta)\right)}\mathcal{N}\big(\bar{E}^n(\theta)\big) \in \mathcal{N}(H(\theta)) \quad j \geq p_l, \forall l.$$

$\square$

Bibliography

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, N.J., 1979.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, third edition, 1999.

C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, March 2004.

K. J. Åström. *Introduction to Stochastic Control Theory*. Mathematics in Science and Engineering. Academic Press, New York and London, 1970.

K. J. Åström and B. Wittenmark. *Computer Controlled Systems, Theory and Design*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, N.J., 1984.

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000.

V. M. Becerra, P. D. Roberts, and G. W. Griffiths. Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9:267–281, 2001.

D. J. Bender and A. J. Laub. The linear-quadratic optimal regulator for descriptor systems. *IEEE Transactions on Automatic Control*, AC-32(8):672–688, August 1987.

T. Bohlin. *Interactive System Identification: Prospects and Pitfalls*. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics In Applied Mathematics. SIAM, Philadelphia, 1996.

L. Chisci and G. Zappa. Square-root Kalman filtering of descriptor systems. *Systems & Control Letters*, 19(4):325–334, October 1992.

D. Cobb. Controllability, observability, and duality in singular systems. *IEEE Transactions on Automatic Control*, AC-29(12):1076–1082, December 1984.

L. Dai. State estimation schemes for singular systems. In *Preprints of the 10th IFAC World Congress, Munich, Germany*, volume 9, pages 211–215, 1987.

L. Dai. Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Transactions on Automatic Control*, 34(10):1105–1108, October 1989a.

L. Dai. *Singular Control Systems*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, New York, 1989b.

M. Darouach, M. Zasadzinski, and D. Mehdi. State estimation of stochastic singular linear systems. *International Journal of Systems Science*, 24(2):345–354, 1993.

M. Darouach, M. Boutayeb, and M. Zasadzinski. Kalman filtering for continuous descriptor systems. In *Proceedings of the American Control Conference*, pages 2108–2112, Albuquerque, New Mexico, June 1997. AACC.

Z. L. Deng and Y. M. Liu. Descriptor kalman filtering. *International Journal of Systems Science*, 30(11):1205–1212, 1999.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in Practice*. Springer-Verlag, New York, 2001.

I. S. Duff and J. K. Reid. An implementation of Tarjan's algorithm for the block triangularization of a matrix. *ACM Transactions on Mathematical Software*, 4(2):137–147, June 1978.

P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE, New York, 2004.

F. R. Gantmacher. *The Theory of Matrices*, volume 2. Chelsea Publishing Company, New York, 1960.

M. Gerdin. Computation of a canonical form for linear differential-algebraic equations. In *Proceedings of Reglermöte 2004*, Göteborg, Sweden, May 2004.

M. Gerdin. Local identifiability and observability of nonlinear differential-algebraic equations. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006a.

M. Gerdin. Using DAE solvers to examine local identifiability for linear and nonlinear systems. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006b.

M. Gerdin and T. Glad. On identifiability of object-oriented models. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.

M. Gerdin and J. Sjöberg. Nonlinear stochastic differential-algebraic equations with application to particle filtering. In *Proceedings of 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006. Accepted for publication.

M. Gerdin, T. Glad, and L. Ljung. Parameter estimation in linear differential-algebraic equations. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1530–1535, Rotterdam, the Netherlands, August 2003.

M. Gerdin, T. Glad, and L. Ljung. Well-posedness of filtering problems for stochastic linear DAE models. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*, pages 350–355, Seville, Spain, December 2005.

M. Gerdin, T. B. Schön, T. Glad, F. Gustafsson, and L. Ljung. On parameter and state estimation for linear differential-algebraic equations. *Automatica*, 2006. To appear.

A. Germani, C. Manes, and P. Palumbo. Kalman-Bucy filtering for singular stochastic differential systems. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, July 2002.

T. Glad and L. Ljung. *Control Theory, Multivariable and Nonlinear Methods*. Taylor and Francis, New York, 2000.

G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore and London, third edition, 1996.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113, April 1993.

S. Graebe. *Theory and Implementation of Gray Box Identification*. PhD thesis, Automatic Control, Royal Institute of Technology, Stockhom, Sweden, 1990.

F. Gustafsson. *Adaptive Filering and Change Detection*. John Wiley & Sons, Ltd, Chichister, Weinheim, New York, Brisbane, Singapore, Toronto, 2000.

R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, AC-22(5):728–740, October 1977.

A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin, Heidelberg, second edition, 1989.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

B. Kågström. A perturbation analysis of the generalized Sylvester equation. *Siam Journal on Matrix Analysis and Applications*, 15(4):1045–1060, October 1994.

T. Kailath. *Linear Systems*. Information and Systems Sciences Series. Prentice Hall, Englewood Cliffs, N.J., 1980.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall Information and System Sciences Series. Prentice Hall, Upper Saddle River, N.J., 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(Series D):35–45, 1960.

L. Kronecker. Algebraische Reduction der Schaaren bilinearer Formen. *Sitzungsberichte der Königlich Preussischen Akadademie der Wissenschaften zu Berlin*, pages 1225–1237, 1890.

V. Kučera. Stationary LQG control of singular systems. *IEEE Transactions on Automatic Control*, AC-31(1):31–39, January 1986.

P. Kunkel and V. Mehrmann. Analysis of over- and underdetermined nonlinear differential-algebraic systems with application to nonlinear control problems. *Mathematics of Control, Signals, and Systems*, 14(3):233–256, 2001.

P. Kunkel and V. Mehrmann. Index reduction for differential-algebraic equations by minimal extension. *Zeitschrift für Angewandte Mathematik und Mechanik*, 84(9):579–597, July 2004.

P. Kunkel and V. Mehrmann. Characterization of classes of singular linear differential-algebraic equations. *Electronic Journal of Linear Algebra*, 13:359–386, November 2005.

P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Zürich, 2006.

P. Kunkel and V. Mehrmann. Canonical forms for linear differential-algebraic equations with variable coefficients. *Journal of Computational and Applied Mathematics*, 56(3): 225–251, December 1994.

F. L. Lewis. A survey of linear singular systems. *Circuits Systems and Signal Processing*, 5(1):3–36, 1986.

L. Ljung. *System Identification Toolbox for use with* MATLAB*: User's Guide. Version 6*. The MathWorks, Inc, Natick, MA, 2006.

L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, AC-24(1):36–50, February 1979.

L. Ljung. *System Identification: Theory for the User*. Information and System Sciences Series. Prentice Hall PTR, Upper Saddle River, N.J., second edition, 1999.

L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, February 1994.

D. G. Luenberger. Time-invariant descriptor systems. *Automatica*, 14:473–480, 1978.

S. E. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific Computing*, 14(3):1064–8275, May 1993.

S. E. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with Modelica. *Control Engineering Practice*, 6:501–510, 1998.

P. C. Müller. Descriptor systems: Analysis and control design. *SACTA*, 3(3):181–195, 2000.

H. Nijmeijer and A. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag, New York, 1990.

R. Nikoukhah, S. L. Campbell, and F. Delebecque. Kalman filtering for general discrete-time LTI systems. In *Proceedings of the 37th IEEE Conference on Decision & Control. Tampa, Florida USA.*, pages 2886–2891. IEEE, December 1998.

R. Nikoukhah, S. L. Campbell, and F. Delebecque. Kalman filtering for general discrete-time linear systems. *IEEE Transactions on Automatic Control*, 44(10):1829–1839, October 1999.

C. C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, March 1988.

A. Papoulis. *Signal Analysis.* McGraw-Hill, 1977.

P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, Pasadena, California, 2000.

P. A. Parrilo and L. Ljung. Initialization of physical parameter estimates. In *Proceedings of the 13th IFAC symposium on system identification*, pages 1524–1529, Rotterdam, the Netherlands, August 2003.

R. Pintelon and J. Schoukens. *System Identification: A frequency domain approach.* IEEE Press, New York, 2001.

J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach.* Number 26 in Texts in Applied Mathematics. Springer-Verlag, New York, 1998.

S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. *SOSTOOLS, Sum of squares optimization toolbox for* MATLAB*, User's guide, Version 2.00*, 2004. Available at `http://www.cds.caltech.edu/sostools`.

G. Reißig, W. S. Martinson, and P. I. Barton. Differential-algebraic equations of index 1 may have an arbitrarily high structural index. *SIAM Journal on Scientific Computing*, 21(6):1987–1990, 2000.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: particle filters for tracking applications.* Artech House, Boston, Mass., London, 2004.

J. F. Ritt. *Differential Algebra*. Dover, New York, 1966.

H. H. Rosenbrock. *State-Space and Multivariable Theory*. John Wiley & Sons, Inc., New York, 1970.

W. J. Rugh. *Linear System Theory*. Prentice Hall, Upper Saddle River, N.J., 1996.

O. Schein and G. Denk. Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *Journal of Computational and Applied Mathematics*, 100(1):77–92, November 1998.

K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publishers, Dordrecht, 2002.

T. Schön and F. Gustafsson. Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC symposium on system identification*, pages 1287–1292, September 2003.

T. Schön, M. Gerdin, T. Glad, and F. Gustafsson. A modeling and filtering framework for linear differential-algebraic equations. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 892–897, Maui, Hawaii, USA, December 2003.

T. B. Schön. *Estimation of Nonlinear Systems: Theory and Applications*. PhD thesis, Linköping University, February 2006.

V. Sima. *Algorithms for linear-quadratic optimization*. Dekker, New York, 1996.

M. Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer, Boston, Mass., 2001.

P. van Overschee and B. De Moor. *Subspace Identification for Linear Systems*. Kluwer Academic Publishers, Boston, London, Dordrecht, 1996.

A. Varga. Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *Prepr. of 2nd IFAC Workshop on System Structure and Control, Prague, Czechoslovakia*, pages 392–395, 1992.

E. Walter. *Identifiability of State Space Models with Applications to Transformation Systems*, volume 46 of *Lecture Notes in Biomathematics*. Springer-Verlag, Berlin, Heidelberg, New York, 1982.

K. Weierstrass. Zur Theorie der bilinearen und quadratischen Formen. *Monatsberichte der Königlich Preussischen Akadademie der Wissenschaften zu Berlin*, pages 310–338, 1868.

R. Winkler. Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of Computational and Applied Mathematics*, 163(2):435–463, February 2004.

E. Wong and B. Hajek. *Stochastic Processes in Engineering Systems*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

L. L. Xie and L. Ljung. Estimate physical parameters by black-box modeling. In *Proceedings of the 21st Chinese Control Conference*, pages 673–677, August 2002.

Index

autocovariance function, 47

Brownian motion, 47

cokernel, 39
corange, 39
corank, 18
covariance function, 47

DAE
    linear, 24
    linear time-varying, 38
    nonlinear, 13
    regular, 23, 25, 41, 44
    sampling, 36
    solvable, 15
    solver, 41
    state-space form, 33
derivative array, 16
difference-algebraic equation, 43
differential algebra, 59
differential-algebraic equation, *see* DAE
Dymola, 8

frequency domain identification, 57
    DAE, 157

Gaussian process, 47
generalized eigenvalue, 174

identifiability, 58
    DAE, 87, 105, 119
implementation, 77, 173
impulse controllability, 136
index
    differential, 14
    strangeness, 19
initial condition
    consistent, 15

kernel, 39
Kunkel and Mehrmann, 15

LAPACK, 173

maximum likelihood method, 57
minimally structurally singular, 123
model, 7
    component-based, 7
    deterministic, 8
    gray-box, 55
    stochastic, 12, 46
Modelica, 7

nilpotent, 197
null-space, 39

oblique projection, 153
observability, 61

DAE, 87
observability indeces, 99
OpenModelica, 8

Pantelides's algorithm, 41
parameter estimation, *see* system identification
particle filter, 73
pole excess, 68
prediction error method, 55

range, 39
regularity, *see* DAE, regular
row degree, 140
row reduced, 140

SDAE
linear, 133
nonlinear, 65
sampling, 144
shuffle algorithm, 31
spectral density, 48
spectrum, 48
state estimation
linear DAE, 163
nonlinear DAE, 73
well-posed problem, 164, 167
state-space model, 11
stationary process, 47
stochastic process, 47
sum of squares optimization, 187
SVD coordinate system, 32
system identification, 55
initialization, 180
linear DAE, 151
nonlinear DAE, 82
well-posed problem, 152, 159

U-indistinguishable, 61

Wiener process, 47

zero dynamics, 101

PhD Dissertations
Division of Automatic Control
Linköpings universitet

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.
A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.
B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.
S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.
H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.
E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.
K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.
B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.
S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.
A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.
M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.
K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.
F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.
P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.
T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.
S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.
H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.
I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.
J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.
K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.
T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.
J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.
R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.
P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.
H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.
P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.
J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.
M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.
U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.
A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.
N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.
K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.
M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.
F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.
V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.
M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.
F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.
J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.
J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.
J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.
O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.
R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.
D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.
R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.
J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.
E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.
M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.
T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.
I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.
J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.