

On Parameter and State Estimation for Linear Differential-Algebraic Equations

Markus Gerdin¹, Thomas B. Schön, Torkel Glad, Fredrik Gustafsson, Lennart Ljung

*Automatic Control
Linköping University
SE-581 83 Linköping, Sweden*

Abstract

The current demand for more complex models has initiated a shift away from state-space models towards models described by differential-algebraic equations (DAE). These models arise as the natural product of object-oriented modeling languages, such as Modelica. However, the mathematics of DAE:s is somewhat more involved than the standard state-space theory. The aim of this work is to present a well-posed description of a stochastic differential-algebraic equation and more importantly explain how well-posed estimation problems can be formed. We will consider both the system identification problem and the state estimation problem. Besides providing the necessary theory we will also explain how the procedures can be implemented by means of efficient numerical methods.

Key words: Differential-algebraic equations, Kalman filtering, Grey-box models, Estimation, Parameter estimation, State estimation, Stochastic differential-algebraic equations, Modeling.

1 Identification, Modeling and Stochastic Differential-Algebraic Equations

System identification is about estimating models from observed signals from a system. It is of increasing interest to combine this with physical modeling, that is to use model structures that are founded in physical understanding of the system. This is often called *grey box identification*. The classical way of dealing with this has been to construct state-space models where unknown constants enter as parameters to be estimated. See, among many references, e.g., [16], Section 4.3, [4], [12], and [19]. There are also software packages for identifying such grey box models, both linear and non-linear ones, e.g., [17], [4], [19].

However, today's modeling efforts are no longer focused on state-space models. Demands on modularity and building of complex models from model libraries have favored *object-oriented modeling*. See, e.g., [8], [21]. In

Email addresses: gerdin@isy.liu.se (Markus Gerdin), schon@isy.liu.se (Thomas B. Schön), torkel@isy.liu.se (Torkel Glad), fredrik@isy.liu.se (Fredrik Gustafsson), ljung@isy.liu.se (Lennart Ljung).

¹ Corresponding author. Tel. +46-13-281393. Fax. +46 13 282622.

an object oriented modeling approach, the user does the work by connecting simple models, often by graphical programming. The program collects all the basic model equations and the connection equations and sorts them to be used for efficient simulations (and other applications). It is not intended that the user should be involved in this organization of equations, or even see the result.

It is natural to have the same approach to grey box identification:

- Build the physical model by connecting simple building blocks
- Point to the physical parameters that are unknown in these blocks
- Mark points where it is likely that disturbances (unmeasured inputs) enter
- Mark which external signals are known (measured inputs)
- Declare which signals are measured (outputs) and the measurement accuracies
- Enter the measured signals and let the computer estimate the unknown parameters

but do not bother about dealing with, or even seeing a complete, organized model.

For this it will be essential to work with model represen-

tations that are close to those of object-oriented modeling tools, like MODELICA. These work with *internal variables* which we will collect in a vector $z(t)$, and *external signals*, which will be denoted by \tilde{u} . The equations that describe the basic models and the connections are mathematical relations between these variables and their derivatives. It is sufficient to consider just first order derivatives of z , since higher order derivatives and derivatives of \tilde{u} can be handled by extending the vector z . (This is also how MODELICA treats the equations.) In this paper we shall only consider linear models, which means that any collection of equations can be summarized as

$$E\dot{z}(t) = Fz(t) + G\tilde{u}(t). \quad (1)$$

This is a *differential-algebraic equation* (DAE). It is also known as a *descriptor form* representation of the model. See, e.g., [6], [5] for the general theory around these.

In general, all physical constants that are required to describe the models and the model connections are not known, so the matrices E, F, G will typically contain unknown parameters.

If E in (1) is invertible, the DAE can easily be converted to a regular state-space model. Otherwise, various transformations can be used that bring out the model properties, see e.g. [6], and the appendices of this paper. An essential feature in this is that a DAE may hide implicit differentiations of \tilde{u} .

It is essential to distinguish between two types of external signals \tilde{u} :

- One that corresponds to *measured* inputs, denoted by u . These may be control inputs chosen by the user, or measurable disturbances.
- One that corresponds to *unmeasured* inputs, denoted by w . These are typically disturbance signals, that are known to occur at certain model connections, but are not measurable. Instead they are typically described as stochastic processes.

A DAE which contains external variables w that are modeled as stochastic processes will be called a *Stochastic Differential-Algebraic Equation* (SDAE).

The modeling process thus results in an SDAE, which contains unknown parameters. The identification problem is to estimate these. For that, the measured inputs u will be used together with other measurements of combinations of the internal variables. For this problem a number of questions arise:

- Can a likelihood function for the estimation of the parameters be formulated, taking into account the disturbance signals w , and the statistics of the measurements?

- Is there a guarantee that the implicit differentiations of w that may be hidden in an SDAE do not lead to non-treatable mathematical objects, like differentiated white noise?
- How should algebraic relationships between the variables z be handled when estimating initial conditions?

These are the questions that will be discussed in the current contribution.

2 Problem Formulation

Consider the SDAE

$$E(\theta)\dot{z}(t) = F(\theta)z(t) + G(\theta)u(t) + \sum_{l=1}^{n_w} J_l(\theta)w_l(t, \theta) \quad (2a)$$

$$z(t_0, \theta) = z_0(\theta) \quad (2b)$$

$$\dim z(t) = n \quad (2c)$$

where θ is a vector of unknown parameters which lie in the domain $D_{\mathcal{M}}$ and $w_l(t, \theta)$ is a Gaussian second order stationary process with spectrum

$$\Phi_{w_l}(\omega, \theta) \quad (3)$$

which is rational in ω with pole excess $2p_l$.

This means that

$$\lim_{\omega \rightarrow \infty} \omega^{2p_l} \Phi_{w_l}(\omega, \theta) = C_l(\theta) \\ 0 < C_l(\theta) < \infty \quad \theta \in D_{\mathcal{M}}.$$

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. 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 that $\det(sE - F)$ is not zero for all s . This condition guarantees that a unique solution $z(t)$ exists if there is no noise, which can be realized by calculating the transfer function of the system. See also [6].

An output vector is measured at sampling instants t_k :

$$y(t_k) = H(\theta)z(t_k) + e(t_k) \quad (4)$$

where $e(t_k)$ is a Gaussian random vector with covariance matrix $R_2(k)$, such that $e(t_k)$ and $e(t_s)$ are independent for $k \neq s$ and also independent of all the processes w_l .

The problem treated in this paper is to estimate the unknown parameters θ using $u(t)$ and $y(t)$. As mentioned

earlier, problems might arise with differentiated noise or with elements of the internal variables $z(t)$ being equal to white noise (which has infinite variance). It must therefore be required that the model structure (2) is well-defined:

Definition 1 Let $z(t, \theta)$ be defined as the solution to (2) for a $\theta \in D_{\mathcal{M}}$. The problem to estimate θ from knowledge of $u(t), t \in [t_0, T]$ and $y(t_k), k = 1, \dots, N; t_k \in [t_0, T]$ is **well-posed** if $H(\theta)z(t_k, \theta)$ has finite variance for all $\theta \in D_{\mathcal{M}}$.

Remark: The initial value $z_0(\theta)$ may not be chosen freely, when computing $z(t, \theta)$. See remark 4 in the next section. The possibly conflicting values in $z_0(\theta)$ will be ignored, and actually have no consequence for the computation of $z(t, \theta)$ for $t > t_0$.

For a well-posed estimation problem the likelihood function can be computed:

$$L(y(t_1), \dots, y(t_N); \theta) \quad (5)$$

which is the value of the joint probability density function for the random vectors $y(t_k)$ at the actual observations. This will be discussed in Section 5.

3 Main result

The main result of this contribution 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

$$\mathcal{R}(A) \text{ and } \mathcal{N}(A)$$

respectively. Furthermore, the following definition of an oblique projection will be used.

Definition 2 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

$$A/_B C \triangleq \begin{pmatrix} 0 & \bar{C} \end{pmatrix} \begin{pmatrix} \bar{B} & \bar{C} \end{pmatrix}^{-1} A. \quad (6)$$

Note that the projection is independent of the choice of bases for B and C .

This definition basically follows the definition in [22, 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 in [22] is not necessary here.

The main result can now be formulated as follows:

Theorem 3 Consider the model (2). Let $\lambda(\theta)$ be a scalar such that $\lambda(\theta)E(\theta) + F(\theta)$ is invertible. Let

$$\bar{E}(\theta) = (\lambda(\theta)E(\theta) + F(\theta))^{-1}E(\theta). \quad (7)$$

Then the estimation problem (2)–(4) is well-posed if and only if

$$\begin{aligned} \left[\bar{E}^j(\theta)(\lambda(\theta)E(\theta) + F(\theta))^{-1}J_l(\theta) \right] /_{\mathcal{R}(\bar{E}^n(\theta))} \mathcal{N}(\bar{E}^n(\theta)) \\ \in \mathcal{N}(H(\theta)) \quad j \geq p_l, \forall l. \quad (8) \end{aligned}$$

PROOF. See Appendix A.

Remark 1: If $\lambda E(\theta) + F(\theta)$ is singular for all λ at some $\theta \in D_{\mathcal{M}}$, the DAE (1) is singular, which means either that the DAE is not solvable, or that a part of z is not uniquely determined by the DAE. See further [6].

Remark 2: 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 (8) is equivalent to well-posedness for every $\lambda(\theta)$ with invertible $\lambda(\theta)E(\theta) + F(\theta)$.

Remark 3: There is a smallest m for which

$$\mathcal{N}(\bar{E}(\theta)^l) = \mathcal{N}(\bar{E}(\theta)^m), \quad \forall l \geq m.$$

This number m is the *index* of the SDAE (2).

Remark 4: $z(t, \theta), t > t_0$, and the likelihood function depend on $z_0(\theta)$ only in terms of

$$z_0(\theta) /_{\mathcal{N}(\bar{E}^n(\theta))} \mathcal{R}(\bar{E}^n(\theta)). \quad (9)$$

The part of $z_0(\theta)$ that does not fulfill (9) cannot be chosen freely, but is of no consequence for the estimation problem. See Section 5.

4 Integrating Sampling

It may happen that a selected output z_k has infinite instantaneous variance. This happens when condition (8) is violated. This could be handled by integrating sampling. This is best illustrated by an example: Let the SDAE be

$$\dot{z}_1(t) = -2z_1(t) + v_1(t) \quad (10a)$$

$$0 = -z_2(t) + v_1(t) \quad (10b)$$

$$y(t) = z_1(t) + z_2(t) + v_2(t) \quad (10c)$$

where $v_l(t)$ are continuous-time white noises. This is not a well-posed problem since z_2 has infinite variance. A common way to handle this would be to apply integrating sampling to y (see e.g., [2]). This would give finite

variances to the sampled values $y(t_k)$, and it would also introduce correlation between the “process noise” v_1 and the measurement noise v_2 .

A more 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

$$z_3(t) = \frac{1}{0.01p + 1} (z_1(t) + z_2(t)).$$

Including this new variable in the SDAE gives

$$\begin{aligned} \dot{z}_1(t) &= -2z_1(t) + v_1(t) \\ \dot{z}_3(t) &= -100z_3(t) + 100z_1(t) + 100v_1(t) \\ 0 &= -z_2(t) + v_1(t) \end{aligned}$$

with the sampled measurements

$$y(t_k) = z_3(t_k) + e(t_k).$$

This is a well-posed problem with no correlation between $v_1(t)$ and $e(t_k)$.

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., Section 7.4 in [16].) 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 (which is possible, since they have rational spectral densities), and transforming the SDAE equations (2)–(4) to standard form, see (B.14)–(B.17), we obtain the following representation of $y(t_k)$:

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + L(\theta)v(t) \quad (11a)$$

$$y(t_k) = C(\theta)x(t_k) \quad (11b)$$

$$+ \sum_{l=1}^m \left(D_l(\theta) \frac{d^{l-1}}{dt^{l-1}} u(t_k) \right) + e(t_k)$$

$$v(t) = \begin{bmatrix} v_1(t) & v_2(t) & \dots & v_{n_v}(t) \end{bmatrix}^T \quad (11c)$$

$$E v(t) v^T(s) = R_1 \delta(t-s) \quad (11d)$$

$$E e(t_k) e^T(t_s) = R_2(k) \delta_{t_k, t_s} \quad (11e)$$

This is a standard linear prediction problem with continuous-time dynamics and continuous-time white

noise and discrete-time measurements. The Kalman filter equations for this are given, e.g., in [13], and they define the one-step ahead predicted outputs $\hat{y}(t_k|\theta)$ and the prediction error variances $\Lambda(t_k, \theta)$. With Gaussian disturbances, we obtain in the usual way the log-likelihood function

$$\begin{aligned} V_N(\theta) &= \frac{1}{2} \sum_{k=1}^N (y(t_k) - \hat{y}(t_k|\theta))^T \Lambda^{-1}(t_k, \theta) \\ &\quad \times (y(t_k) - \hat{y}(t_k|\theta)) + \log \det \Lambda(t_k, \theta). \end{aligned} \quad (12)$$

The parameter estimates are then computed as

$$\hat{\theta}_{ML} = \arg \min_{\theta} V_N(\theta). \quad (13)$$

In practice, the important question of how the state-space description should be computed remains. As discussed in Appendix C, the form (11) 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 (12) using a Gauss-Newton search. For each parameter value θ that the Gauss-Newton algorithm needs, the transformed system (11) 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 (A.3) we get that

$$z(t_0) = \begin{bmatrix} T_1(\theta) & T_2(\theta) \end{bmatrix} \begin{bmatrix} x_s(t_0) \\ x_a(t_0) \end{bmatrix}. \quad (14)$$

It is also obvious from the transformed system equations (A.4a) and (A.7) 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 $z(t_0)$ that can be parameterized is thus

$$z(t_0)/\mathcal{R}(T_2)\mathcal{R}(T_1) = z(t_0)/_{\mathcal{N}(\bar{E}^n(\theta))}\mathcal{R}(\bar{E}^n(\theta))$$

where $\bar{E}(\theta)$ is the matrix defined in (7). Note that since x_a is determined by (A.7), any initial conditions that are specified for x_a can be ignored in the identification procedure since they do not affect the likelihood function.

6 State Estimation

In many applications it is useful to estimate variables that are not measured. The standard method to estimate such variables for state-space systems is the Kalman filter. In this section it will be discussed how the Kalman filter can be used to estimate the internal variables $z(t)$ of an SDAE. As for the identification case, the problem must be well-posed. The results follow directly from the earlier discussion, so we will be rather brief. Consider the SDAE

$$E\dot{z}(t) = Fz(t) + Gu(t) + \sum_{l=1}^{n_w} J_l w_l(t) \quad (15a)$$

$$z(t_0) = z_0 \quad (15b)$$

$$\dim z(t) = n \quad (15c)$$

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 $2p_l$. The input $u(t)$ is known for all $t \in [t_0, T]$, and is differentiable a sufficient number of times. An output vector is measured at sampling instants t_k :

$$y(t_k) = Hz(t_k) + e(t_k) \quad (16)$$

where $e(t_k)$ is a Gaussian random vector with covariance matrix $R_2(k)$, such that $e(t_k)$ and $e(t_s)$ are independent for $k \neq s$ and also independent of all the processes w_l .

As for the parameter estimation problem, it must be required that $y(t_k)$ has finite variance. For the estimation problem to make sense, it must also be required that the part of $z(t)$ that is to be estimated has finite variance. The part of $z(t)$ that is to be estimated will be written as $Mz(t)$ for some constant matrix M .

Definition 4 Let $z(t)$ be defined as the solution to (15). The problem to estimate $Mz(t)$ from knowledge of $u(t), t \in [t_0, T]$ and $y(t_k), k = 1, \dots, N; t_k \in [t_0, T]$ is **well-posed** if $Hz(t_k)$ and $Mz(t_k)$ have finite variance.

Remark: As discussed earlier, the initial value z_0 may not be chosen freely, but the possibly conflicting values have no consequence for the computation of $z(t)$ for $t > t_0$.

As for the parameter estimation problem, it is possible to examine if a problem is well-posed using certain subspaces:

Theorem 5 Consider (15)–(16). Let λ be a scalar such that $(\lambda E + F)$ is invertible. Let

$$\bar{E} = (\lambda E + F)^{-1}E. \quad (17)$$

Then the estimation problem (15)–(16) is well-posed if and only if

$$[\bar{E}^j(\lambda E + F)^{-1}J_l] / \mathcal{R}(\bar{E}^n) \mathcal{N}(\bar{E}^n) \in \mathcal{N}(H) \quad j \geq p_l, \forall l \quad (18a)$$

$$[\bar{E}^j(\lambda E + F)^{-1}J_l] / \mathcal{R}(\bar{E}^n) \mathcal{N}(\bar{E}^n) \in \mathcal{N}(M) \quad j \geq p_l, \forall l. \quad (18b)$$

PROOF. This result follows directly from Theorem 3.

As discussed previously, the disturbances $w_l(t)$ can be written as outputs from linear filters, driven by white noise v_l . Transforming the SDAE to standard form, see (B.14)–(B.17), gives the following representation of $y(t_k)$ and $z(t_k)$. The equation for $Mz(t_k)$ is not explicitly given in the appendix, but it can be treated as a second measurement without measurement noise.

$$\dot{x}(t) = Ax(t) + Bu(t) + Lv(t) \quad (19a)$$

$$Mz(t) = \bar{C}x(t) + \sum_{l=1}^m \left(\bar{D}_l \frac{d^{l-1}}{dt^{l-1}} u(t) \right) \quad (19b)$$

$$y(t_k) = Cx(t_k) + \sum_{l=1}^m \left(D_l \frac{d^{l-1}}{dt^{l-1}} u(t_k) \right) + e(t_k) \quad (19c)$$

$$v(t) = [v_1(t) \ v_2(t) \ \dots \ v_{n_v}(t)]^T \quad (19d)$$

$$Ev(t)v^T(s) = R_1\delta(t-s) \quad (19e)$$

$$Ee(t_k)e^T(s_k) = R_2(k)\delta_{t_k, s_k} \quad (19f)$$

As noted earlier, this filtering problem can be solved using the Kalman filter (e.g., [13]).

The problem of estimating internal variables in DAE:s and modeling SDAE:s has to some extent been discussed by other authors. In [18], it is guaranteed that the noise is not differentiated by assuming that the system is *index 1* (see, e.g., [5]). The assumption that the system is index 1 is more restrictive than is necessary, and rules out some applications such as many mechanics systems. [18] also notes that some internal variables actually may be *generalized stochastic processes*, that is, equal to a white noise process. [24] makes the same assumption as [18], but also treats a class of nonlinear SDAE.

In [7] index 1 is assumed 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 [15], the original system specification may specify derivatives of white noise, but a controller is designed that removes any derivatives. In [11] the restrictive assumption that $\mathcal{R}(FG) \subseteq \mathcal{R}(E)$ guarantees that no derivatives appear, although this is

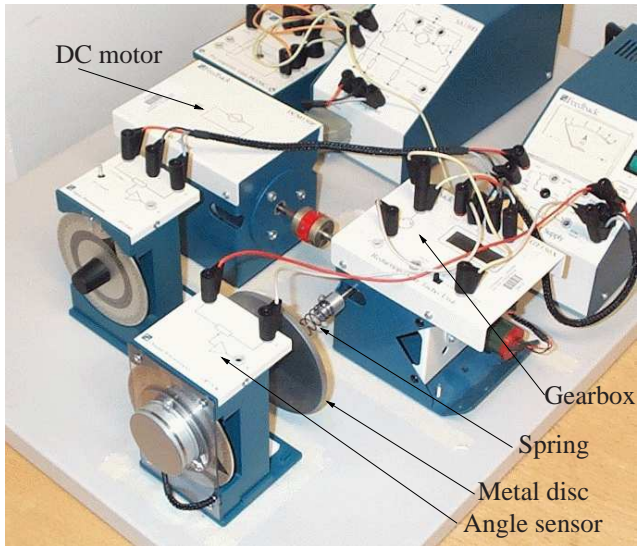


Fig. 1. The examined process.

not stated explicitly. In [3] nonlinear semi-explicit SDAE (see, e.g., [5]) are discussed. Here well-posedness is guaranteed by only adding noise to the state-space part of the system. In [20] a transformation to a standard form is used to study when the filter problem is well-defined. Finally, in [10] the state estimation approach described in this section is discussed in more detail.

7 Application Example

In this section the algorithms examined earlier in the paper are exemplified on a physical system. The system setup examined is a DC-motor connected to a heavy metal disc through a gearbox and a spring, see Figure 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

$$E(\theta)\dot{z}(t) = F(\theta)z(t) + G(\theta)u(t) \quad (20a)$$

$$y(t) = H(\theta)z(t). \quad (20b)$$

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 grey-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, [17] 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 A, B, C, D, K, x_0 . This means that this m-file will call the transformation routines described in Appendix C, 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 descriptor 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 (2)–(4) for a certain parameter value, and the function `StandardForm` computes a corresponding state-space description using LAPACK (see Appendix C). 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.

8 Conclusions

The main result of this contribution is Theorem 3, where we provide necessary and sufficient conditions for an estimation problem, formed from a stochastic differential-algebraic equation, to be well-posed. Furthermore, we have provided a solid motivation of the meaning of a well-posed stochastic differential-algebraic equation. The application of Theorem 3 to solve the system identification and state estimation problems was also described,

not only theoretically, we also provide guidelines for an efficient implementation of the results using numerical methods. Finally, we presented an application example.

9 Acknowledgements

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

A Proof of Theorem 3

In this appendix Theorem 3 is proved. Recall that $\lambda(\theta)$ is a scalar such that $\lambda(\theta)E(\theta) + F(\theta)$ is invertible and

$$\bar{E}(\theta) = (\lambda(\theta)E(\theta) + F(\theta))^{-1}E(\theta). \quad (\text{A.1})$$

First we will prove two propositions:

Proposition 6 Consider the SDAE (2) with the matrix $\bar{E}(\theta)$ transformed into Jordan form:

$$\bar{E}(\theta) = \begin{bmatrix} T_1(\theta) & T_2(\theta) \end{bmatrix} \begin{bmatrix} E_s(\theta) & 0 \\ 0 & N(\theta) \end{bmatrix} \begin{bmatrix} T_1(\theta) & T_2(\theta) \end{bmatrix}^{-1} \quad (\text{A.2})$$

where the zero eigenvalues are sorted to the lower right so that E_s is invertible and N is nilpotent of order m .

Then the transformation

$$z = \underbrace{\begin{bmatrix} T_1(\theta) & T_2(\theta) \end{bmatrix}}_{T(\theta)} \underbrace{\begin{bmatrix} x_s \\ x_a \end{bmatrix}}_x \quad (\text{A.3})$$

gives a system description of the form

$$E_s(\theta)\dot{x}_s = (I - \lambda(\theta)E_s(\theta))x_s + G_s(\theta)u + \sum_{l=1}^{n_w} J_{l,s}(\theta)w_l(\theta) \quad (\text{A.4a})$$

$$N(\theta)\dot{x}_a = (I - \lambda(\theta)N(\theta))x_a + G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta) \quad (\text{A.4b})$$

where

$$\begin{bmatrix} J_{l,s}(\theta) \\ J_{l,a}(\theta) \end{bmatrix} = T^{-1}(\theta)(\lambda(\theta)E(\theta) + F(\theta))^{-1}J_l(\theta) \quad (\text{A.5})$$

$$\begin{bmatrix} G_s(\theta) \\ G_a(\theta) \end{bmatrix} = T^{-1}(\theta)(\lambda(\theta)E(\theta) + F(\theta))^{-1}G(\theta). \quad (\text{A.6})$$

PROOF. Adding $\lambda(\theta)E(\theta)z$ to each side of Equation (2a) and then multiplying from the left with $(\lambda E(\theta) + F(\theta))^{-1}$ gives

$$\begin{aligned} \bar{E}(\theta)(\dot{z} + \lambda(\theta)z) &= z + (\lambda(\theta)E(\theta) + F(\theta))^{-1} \\ &\times \left(G(\theta)u + \sum_{l=1}^{n_w} J_l(\theta)w_l(\theta) \right). \end{aligned}$$

Substituting $z = Tx$ and multiplying from the left with T^{-1} gives

$$\begin{aligned} T^{-1}\bar{E}(\theta)T(\dot{x} + \lambda x) &= x + T^{-1}(\lambda E(\theta) + F(\theta))^{-1} \\ &\times \left(G(\theta)u + \sum_{l=1}^{n_w} J_l(\theta)w_l(\theta) \right) \end{aligned}$$

which is the desired form.

Proposition 7 The auxiliary variables x_a can be solved from (A.4b) to give

$$\begin{aligned} x_a &= - \left(I + \left(\frac{d}{dt} + \lambda(\theta) \right) N(\theta) + \dots + \right. \\ &\quad \left. \left(\frac{d}{dt} + \lambda(\theta) \right)^{m-1} N^{m-1}(\theta) \right) \times \\ &\quad \left(G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta) \right) \quad (\text{A.7}) \end{aligned}$$

PROOF. Writing (A.4b) as

$$\begin{aligned} x_a &= N(\theta) \left(\frac{d}{dt} + \lambda(\theta) \right) x_a \\ &\quad - \left(G_a(\theta)u + \sum_{l=1}^{n_w} J_{l,a}(\theta)w_l(\theta) \right) \quad (\text{A.8}) \end{aligned}$$

and successively differentiating and multiplying by $N(\theta)$ gives (omitting dependence on θ)

$$\begin{aligned} N \left(\frac{d}{dt} + \lambda \right) x_a &= N^2 \left(\frac{d}{dt} + \lambda \right)^2 x_a \\ &\quad - N \left(\frac{d}{dt} + \lambda \right) \left(G_a u + \sum_{l=1}^{n_w} J_{l,a} w_l(\theta) \right) \\ &\quad \vdots \\ N^{m-1} \left(\frac{d}{dt} + \lambda \right)^{m-1} x_a &= \\ &\quad - N^{m-1} \left(\frac{d}{dt} + \lambda \right)^{m-1} \left(G_a u + \sum_{l=1}^{n_w} J_{l,a} w_l \right) \end{aligned}$$

where we have used $N^m = 0$ in the last equation. A successive substitution from these equations into (A.8) then gives (A.7).

We now prove the main result, Theorem 3.

PROOF. Transforming the system into the form (A.4) we see that the equation for x_s can be interpreted as the stochastic differential equation

$$dx_s = (E_s^{-1}(\theta) - \lambda(\theta)I)x_s dt + E_s^{-1}(\theta)G_s(\theta)u dt + E_s^{-1}(\theta) \sum_{l=1}^{n_w} J_{l,s}(\theta)dw_l \quad (\text{A.9})$$

so x_s has finite variance. Since

$$H(\theta)z = H(\theta)T_1(\theta)x_s + H(\theta)T_2(\theta)x_a$$

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 $2p_l$. (A.7) thus gives that $H(\theta)T_2(\theta)x_a$ has finite variance if and only if

$$H(\theta)T_2(\theta)N^j(\theta)J_{l,a}(\theta) = 0 \quad j \geq p_l, \forall l.$$

By using the notation $[\cdot]_{\mathcal{X}} \mathcal{Y}$ for the oblique projection on the space \mathcal{Y} along the space \mathcal{X} and $\mathcal{R}(A)$ for the space spanned by the columns of the matrix A , this condition can be written as (omitting dependence on θ)

$$\begin{aligned} 0 &= HT_2 N^j J_{l,a} \\ &= H [T_1 E_s^j J_{l,s} + T_2 N^j J_{l,a}] /_{\mathcal{R}(T_1)} \mathcal{R}(T_2) \\ &= H \left[\begin{pmatrix} T_1 & T_2 \end{pmatrix} \begin{pmatrix} E_s^j & 0 \\ 0 & N^j \end{pmatrix} \begin{pmatrix} J_{l,s} \\ J_{l,a} \end{pmatrix} \right] /_{\mathcal{R}(T_1)} \mathcal{R}(T_2) \\ &= H [\bar{E}^j (\lambda E + F)^{-1} J_l] /_{\mathcal{R}(T_1)} \mathcal{R}(T_2). \end{aligned}$$

Since $E_s(\theta)$ is invertible and $N(\theta)$ is nilpotent, (A.2) gives that $\mathcal{R}(T_2(\theta)) = \mathcal{N}(\bar{E}^n(\theta))$ and that $\mathcal{R}(T_1(\theta)) = \mathcal{R}(\bar{E}^n(\theta))$, so the condition can also be written

$$\begin{aligned} & \left[\bar{E}^j(\theta) (\lambda(\theta)E(\theta) + F(\theta))^{-1} J_l(\theta) \right] /_{\mathcal{R}(\bar{E}^n(\theta))} \mathcal{N}(\bar{E}^n(\theta)) \\ & \in \mathcal{N}(H(\theta)) \quad j \geq p_l, \forall l. \end{aligned}$$

B Standard Form

To implement estimation procedures, it is useful to convert an SDAE into a state-space-like form. One method to do this will be presented in this appendix. It will be

assumed that the corresponding estimation problem is well-posed.

Consider the original SDAE

$$E(\theta)\dot{z}(t) = F(\theta)z(t) + G(\theta)u(t) + \sum_{l=1}^{n_w} J_l(\theta)w_l(t, \theta) \quad (\text{B.1})$$

$$z(t_0, \theta) = z_0(\theta) \quad (\text{B.2})$$

$$\dim z(t) = n \quad (\text{B.3})$$

where $w_l(t, \theta)$ is a Gaussian second order stationary process with spectrum $\Phi_{w_l}(\omega, \theta)$ which is rational in ω with pole excess $2p_l$. An output vector is measured at sampling instants t_k :

$$y(t_k) = H(\theta)z(t_k) + e(t_k) \quad (\text{B.4})$$

where $e(t_k)$ is a Gaussian random vector with covariance matrix $R_2(k)$, such that $e(t_k)$ and $e(t_s)$ are independent for $k \neq s$ and also independent of all the processes w_l .

Since the disturbances $w_l(t, \theta)$ have rational spectra, it is possible to write them as outputs from linear filters driven by white noise, so that

$$\dot{z}_w(t) = A_w(\theta)z_w(t) + B_w(\theta)v(t) \quad (\text{B.5a})$$

$$w(t, \theta) = C_w(\theta)z_w(t) + D_w(\theta)v(t) \quad (\text{B.5b})$$

where

$$v(t) = [v_1(t) \cdots v_{n_w}(t)]^T \quad (\text{B.6})$$

is white noise with variance $R_1\delta(t-s)$ and

$$w(t, \theta) = [w_1(t, \theta) \cdots w_{n_w}(t, \theta)]^T. \quad (\text{B.7})$$

By writing

$$J(\theta) = [J_1(\theta) \cdots J_{n_w}(\theta)] \quad (\text{B.8})$$

(B.1), (B.4), and (B.5) can be combined to give

$$\begin{aligned} \begin{bmatrix} E(\theta) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{z}(t) \\ \dot{z}_w(t) \end{bmatrix} &= \begin{bmatrix} F(\theta) & J(\theta)C_w(\theta) \\ 0 & A_w(\theta) \end{bmatrix} \begin{bmatrix} z(t) \\ z_w(t) \end{bmatrix} \\ &+ \begin{bmatrix} G(\theta) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} J(\theta)D_w(\theta) \\ B_w(\theta) \end{bmatrix} v(t) \end{aligned} \quad (\text{B.9a})$$

$$y(t_k) = \begin{bmatrix} H(\theta) & 0 \end{bmatrix} \begin{bmatrix} z(t_k) \\ z_w(t_k) \end{bmatrix} + e(t_k). \quad (\text{B.9b})$$

It is a standard result (e.g., [6]) that there exist non-singular matrices $P(\theta)$ and $Q(\theta)$ such that multiplying (B.9a) from the left with $P(\theta)$ and doing the variable transformation

$$\begin{bmatrix} z(t) \\ z_w(t) \end{bmatrix} = Q(\theta) \begin{bmatrix} x_s(t) \\ x_a(t) \end{bmatrix} \quad (\text{B.10})$$

gives a system of the form

$$\begin{bmatrix} I & 0 \\ 0 & N(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}_s(t) \\ \dot{x}_a(t) \end{bmatrix} = \begin{bmatrix} A(\theta) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_a(t) \end{bmatrix} + \begin{bmatrix} G_s(\theta) \\ G_a(\theta) \end{bmatrix} u(t) + \begin{bmatrix} J_s(\theta) \\ J_a(\theta) \end{bmatrix} v(t) \quad (\text{B.11a})$$

$$y(t_k) = \begin{bmatrix} C_s(\theta) & C_a(\theta) \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_a(t) \end{bmatrix} + e(t_k) \quad (\text{B.11b})$$

where $N(\theta)$ is a nilpotent matrix so that $N^m(\theta) = 0$ for some θ . That this transformation exists can also be realized from, e.g., the Kronecker canonical form for a matrix pencil, see e.g., [9].

Writing the second row of (B.11a) as

$$x_a(t) = N(\theta) \frac{d}{dt} x_a(t) - G_a(\theta) u(t) - J_a(\theta) v(t) \quad (\text{B.12})$$

and successively differentiating and multiplying by $N(\theta)$ gives

$$\begin{aligned} N(\theta) \frac{d}{dt} x_a(t) &= N^2(\theta) \frac{d}{dt} \dot{x}_a(t) \\ &\quad - N(\theta) \frac{d}{dt} (G_a(\theta) u(t) - J_a(\theta) v(t)) \\ &\quad \vdots \\ N^{m-1}(\theta) \frac{d^{m-1}}{dt^{m-1}} x_a(t) &= \\ &\quad - N^{m-1} \frac{d^{m-1}}{dt^{m-1}} (G_a(\theta) u(t) - J_a(\theta) v(t)) \end{aligned}$$

where $N^m(\theta) = 0$ has been used in the last equation. Successively substituting this into (B.12) gives

$$x_a(t) = - \left(I + \frac{d}{dt} N(\theta) + \dots + \frac{d^{m-1}}{dt^{m-1}} N^{m-1}(\theta) \right) \times (G_a(\theta) u(t) + J_a(\theta) v(t)). \quad (\text{B.13})$$

Inserting into (B.11b) gives (omitting dependence on θ)

$$y(t_k) = C_s x_s(t_k) + C_a \times \sum_{l=1}^m \left(\frac{d^{l-1}}{dt^{l-1}} N^{l-1} (G_a u(t_k) + J_a v(t_k)) \right) + e(t_k).$$

If it is assumed that the SDAE forms a well-posed estimation problem, $y(t_k)$ does not depend on white noise, i.e., $v(t)$. This means that $y(t_k)$ can be written as

$$y(t_k) = C_s x_s(t_k) + C_a \sum_{l=1}^m \left(\frac{d^{l-1}}{dt^{l-1}} N^{l-1} G_a u(t_k) \right) + e(t_k).$$

The above discussion gives that an SDAE that forms a well-posed estimation problem can be written in the state-space like form

$$\dot{x}_s(t) = A(\theta) x_s(t) + G_s(\theta) u(t) + J_s(\theta) v(t) \quad (\text{B.14a})$$

$$y(t_k) = C_s(\theta) x_s(t_k) \quad (\text{B.14b})$$

$$+ C_a(\theta) \sum_{l=1}^m \frac{d^{l-1}}{dt^{l-1}} N^{l-1}(\theta) G_a(\theta) u(t_k) + e(t_k)$$

where

$$v(t) = \begin{bmatrix} v_1(t) & \dots & v_{n_v}(t) \end{bmatrix}^T \quad (\text{B.15})$$

is continuous-time white noise signals with variance

$$E v(t) v^T(s) = R_1 \delta(t - s) \quad (\text{B.16})$$

and $e(t_k)$ is a sequence of discrete-time white noise with variance

$$E e(t_k) e^T(s_k) = R_2(k) \delta_{t_k, s_k}. \quad (\text{B.17})$$

We call (B.14)–(B.17) the standard form for an SDAE.

C Numerical Methods

The transformation to (B.11) can be computed numerically with tools from the linear algebra package LAPACK [1]. LAPACK is a 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 by commercial software like MATLAB.

Some ideas related to the method presented in this section for computing the canonical form, have earlier been published in [23]. The presentation here is however more detailed, and uses the software from the freely available LAPACK package.

The computation is performed by first transforming the system to *generalized real Schur form* and then solving a

generalized Sylvester equation as described in the numbered list below.

- (1) Start with an SDAE system:

$$E\dot{z}(t) = Fz(t) + Gu(t) + Jv(t) \quad (\text{C.1a})$$

$$y(t_k) = Hz(t_k) + e(t_k) \quad (\text{C.1b})$$

The goal is to find the transformation $PEQQ^{-1}\dot{z}(t) = PFQQ^{-1}z(t) + PGu(t) + PJv(t)$ that converts it to the form

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} Q^{-1}\dot{z}(t) = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} Q^{-1}z(t) + \begin{bmatrix} G_s \\ G_a \end{bmatrix} u(t) + \begin{bmatrix} J_s \\ J_a \end{bmatrix} v(t) \quad (\text{C.2a})$$

$$y(t_k) = \begin{bmatrix} C_s & C_a \end{bmatrix} Q^{-1}z(t) + e(t_k). \quad (\text{C.2b})$$

- (2) Compute the generalized real Schur form of the matrix pencil $\lambda E - F$ so that

$$P_1(\lambda E - F)Q_1 = \lambda \begin{bmatrix} E_1 & E_2 \\ 0 & E_3 \end{bmatrix} + \begin{bmatrix} F_1 & F_2 \\ 0 & F_3 \end{bmatrix} \quad (\text{C.3})$$

where E_1 , E_3 , F_1 , and F_3 are upper triangular matrices, possibly with some 2×2 blocks on the diagonal corresponding to complex eigenvalues. The diagonal elements should be sorted so that diagonal elements of E_1 contain only non-zero elements and the diagonal elements of E_3 are zero. Note that F_3 will have non-zero diagonal elements since $\det(sE - F) \neq 0$.

This computation can be made with one of the LAPACK commands `dgges` and `sgges`.

- (3) To get from the block triangular form (C.3) to a block diagonal form, solve the generalized Sylvester equation

$$E_1R + LE_3 = -E_2 \quad (\text{C.4a})$$

$$F_1R + LF_3 = -F_2 \quad (\text{C.4b})$$

to get the matrices L and R . The generalized Sylvester equation (C.4) can be solved from the linear equation system

$$\begin{bmatrix} I_n \otimes E_1 & E_3^T \otimes I_m \\ I_n \otimes F_1 & F_3^T \otimes I_m \end{bmatrix} \begin{bmatrix} \text{vec}(R) \\ \text{vec}(L) \end{bmatrix} = \begin{bmatrix} -\text{vec}(E_2) \\ -\text{vec}(F_2) \end{bmatrix},$$

see [14]. Here I_n is an identity matrix with the same size as E_3 and F_3 , I_m is an identity matrix with the same size as E_1 and F_1 , \otimes represents the Kronecker

product and $\text{vec}(X)$ denotes an ordered stack of the columns of a matrix X from left to right starting with the first column. Since this system of equations can be large, efficiency can be gained by using the specialized LAPACK commands `stgsyl` or `dtgsyl` for solving (C.4).

The blocks E_2 and F_2 can now be removed:

$$\begin{bmatrix} I & L \\ 0 & I \end{bmatrix} \begin{bmatrix} E_1 & E_2 \\ 0 & E_3 \end{bmatrix} \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} = \begin{bmatrix} E_1 & E_1R + E_2 + LE_3 \\ 0 & E_3 \end{bmatrix} = \begin{bmatrix} E_1 & 0 \\ 0 & E_3 \end{bmatrix}$$

$$\begin{bmatrix} I & L \\ 0 & I \end{bmatrix} \begin{bmatrix} F_1 & F_2 \\ 0 & F_3 \end{bmatrix} \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} = \begin{bmatrix} F_1 & F_1R + F_2 + LF_3 \\ 0 & F_3 \end{bmatrix} = \begin{bmatrix} F_1 & 0 \\ 0 & F_3 \end{bmatrix}$$

- (4) To summarize, (C.2) is obtained according to

$$\begin{aligned} P &= \begin{bmatrix} E_1^{-1} & 0 \\ 0 & F_3^{-1} \end{bmatrix} \begin{bmatrix} I & L \\ 0 & I \end{bmatrix} P_1 & Q &= Q_1 \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} \\ N &= F_3^{-1}E_3 & A &= E_1^{-1}F_1 \\ \begin{bmatrix} G_s \\ G_a \end{bmatrix} &= PG & \begin{bmatrix} J_s \\ J_a \end{bmatrix} &= PJ. \end{aligned}$$

Note that E_1 and F_3 are invertible since they are upper triangular with non-zero diagonal elements and that N is nilpotent since the diagonal elements of E_3 are zero.

References

- [1] 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.
- [2] K. J. Åström. *Introduction to Stochastic Control Theory*. Mathematics in Science and Engineering. Academic Press, New York and London, 1970.
- [3] 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.
- [4] T. Bohlin. *Interactive System Identification: Prospects and Pitfalls*. Springer-Verlag, Berlin, 1991.
- [5] 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.

- [6] L. Dai. *Singular Control Systems*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, New York, 1989.
- [7] 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.
- [8] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE, New York, 2004.
- [9] F. R. Gantmacher. *The Theory of Matrices*, volume 2. Chelsea Publishing Company, New York, 1960.
- [10] 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*, December 2005. Accepted for publication.
- [11] 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.
- [12] S. Graebe. *Theory and Implementation of Gray Box Identification*. PhD thesis, Automatic Control, Royal Institute of Technology, Stockholm, Sweden, 1990. TRITA-REG 90/3 and 90/6.
- [13] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [14] B. Kågström. A perturbation analysis of the generalized Sylvester equation. *Siam Journal on Matrix Analysis and Applications*, 15(4):1045–1060, October 1994.
- [15] V. Kučera. Stationary LQG control of singular systems. *IEEE Transactions on Automatic Control*, AC-31(1):31–39, January 1986.
- [16] L. Ljung. *System Identification - Theory for the User*. Information and System Sciences Series. Prentice Hall PTR, Upper Saddle River, N.J., second edition, 1999.
- [17] L. Ljung. *System Identification Toolbox for use with MATLAB. Version 6*. The MathWorks, Inc, Natick, MA, sixth edition, 2003.
- [18] 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.
- [19] K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publishers, Dordrecht, 2002.
- [20] 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.
- [21] M. Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer, Boston, Mass., 2001.
- [22] P. van Overschee and B. De Moor. *Subspace Identification for Linear Systems*. Kluwer Academic Publishers, Boston, London, Dordrecht, 1996.
- [23] 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.
- [24] 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.