Parameter identification in linear time invariant descriptor systems

Markus Gerdin
Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: gerdin@isy.liu.se

April 16, 2003

Report no.: LiTH-ISY-R-2514
Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications.
Parameter identification in linear time invariant descriptor systems

Markus Gerdin

April 16, 2003

Abstract

This report treats some basics on parameter identification in linear time invariant descriptor systems, both time continuous and time discrete. It is shown how identification methods for normal state space systems can be used by transforming the descriptor system to state space form. It is also examined how identification can be performed in the frequency domain.

1 Introduction

A linear descriptor system is a system on the form

\[ E\dot{\xi}(t) = J\xi(t) + Ku(t) \]  
\[ y(t) = L\xi(t) \] (1a)

in continuous time and the form

\[ E\xi(t+1) = J\xi(t) + Ku(t) \]  
\[ y(t) = L\xi(t) \] (2a)

in discrete time. Descriptor systems are also called implicit systems, singular systems and, in the time continuous case, differential-algebraic equations (DAEs). \( E, J, K \) and \( L \) are constant matrices, \( \xi(t) \) is a vector of generalized states (note that this is not a true state vector as not all values of \( \xi(t) \) satisfy the equations in the general case) and \( y(t) \) is a measured output. The matrices \( E \) and \( A \) are assumed to be square, furthermore \( E \) is normally not invertible, otherwise the descriptor system can be written on state space form by the variable transformation \( \xi(t) = E\xi(t) \).

If some parameters \( \theta \) of the matrices \( E, J, K \) and \( L \) are unknown, it may be interesting to identify these parameters from the input \( u(t) \) and measurements of the output \( y(t) \). The normal approach in system identification is to find a scalar function \( V(\theta) \) such that it gives an estimation of \( \theta \) when it is minimized over \( \theta \), and then minimize this function [16]. How \( V(\theta) \) should be selected for linear descriptor systems is treated in this report.

Previously, not very much work has been done on parameter identification for descriptor systems, especially not in the linear case. Some works on the nonlinear case are [23, 13, 11].
2 Time continuous descriptor systems

2.1 Basic facts

In this section two three facts about continuous linear descriptor systems are established, namely the solvability of (1), the possibility to put (1) on a normal form and how a discrete time representation of the continuous time descriptor system can be calculated.

2.1.1 Solvability

The solvability of (1) can be said to be that there should exist a unique solution \( \xi(t) \). The solvability turns out to be dependent only on the matrices \( E \) and \( J \). We formalize the solvability with a definition.

**Definition 2.1 (Solvability).** Solvability of the continuous linear descriptor system

\[
E \dot{\xi}(t) - J \xi(t) = K u(t)
\]

is the existence of a unique solution \( \xi(t) \), \( t \geq 0 \), for every infinitely differentiable input \( u(t) \) with \( u(0) = 0 \) and \( \xi(0) = 0 \).

If we restrict ourselves to sequences \( \xi(t) \) and \( u(t) \) that can be Laplace-transformed, we get the following criterion for checking if a time continuous descriptor system is solvable.

**Theorem 2.1.** If only sequences \( \xi(t) \) and \( u(t) \) that can be Laplace-transformed are considered, solvability of (3) is equivalent to that \( \det(sE - J) \neq 0 \) for all \( s \).

**Proof.** First transform (3) to get \( (\xi(0) = 0) \)

\[
(sE - J) \mathcal{L}[\xi(t)] = K \mathcal{L}[u(t)].
\]

Now assume that \( \det(sE - J) \) is not zero for all \( s \). This implies that the inverse exists, and we proceed to multiply (4) from the left with \( (sE - J)^{-1} \) and get

\[
\mathcal{L}[\xi(t)] = (sE - J)^{-1} K \mathcal{L}[u(t)].
\]

Thus \( \xi(t) \) is uniquely determined as the inverse Laplace transform of the right hand side of (5).

On the contrary, assume that \( \det(sE - J) \) is zero for all \( s \). We can then choose a vector \( \alpha(s) \neq 0 \) such that \( (sE - J)\alpha(s) = 0 \). This gives that if \( \mathcal{L}[\xi(t)] \) is a solution of (4), then so is \( \mathcal{L}[\xi(t)] + \alpha(s) \) so a solution is not unique.

Also note that if \( \det(sE - J) \) is zero for all \( s \), a solution \( \xi(t) \) which satisfies (3) may not even exist. The definition of solvability which is presented here is not the same as in for example [4]. Both definitions do however lead to Theorem 2.1.

To our knowledge, solvability is a reasonable requirement to put on a model of a physical systems, and therefore only solvable systems are considered in this report.
2.1.2 The normal form

In this section we extend corollary A.1 in the appendix to include systems with an output \( y \). That is, we examine what the solution of (1) is, assuming that \( sE - J \) is invertible. The result is the following theorem.

**Theorem 2.2.** If \( \det(sE - J) \neq 0 \) for all \( s \), the solution of
\[
E \dot{\xi}(t) = J\xi(t) + Ku(t) \quad (6a)
\]
\[
y(t) = L\xi(t) \quad (6b)
\]
can be described by
\[
\dot{w}_1(t) = Aw_1(t) + Bu(t) \quad (7a)
\]
\[
w_2(t) = \sum_{i=0}^{m-1} (-N)^i Du^{(i)}(t) \quad (7b)
\]
\[
\xi(t) = T^{-1}w(t) \quad (7c)
\]
\[
y(t) = LT^{-1}w(t) \quad (7d)
\]

with notation from appendix A.

**Proof.** The result follows directly from corollary A.1 in the appendix. \( \square \)

Note that the system may depend on derivatives of the input. Similar transformations of linear descriptor systems have been suggested, e.g., in [8, 4].

2.2 Discretization

When doing system identification, the measured data that is to be used to identify the parameters is usually available as sampled data in the time domain. Therefore it is natural to ask what the discrete time representation of a continuous time descriptor system (1) is, and in this section we examine what the exact discrete time description for a time continuous descriptor system is under certain assumptions on the input.

For the derivation of the discrete time representation, we need the following lemma which describes the discrete time representation of normal state space systems.

**Lemma 2.1.** Consider the the state space system
\[
\dot{x}(t) = Ax(t) + Bu(t) \quad (8a)
\]
\[
y(t) = Cx(t) + Du(t). \quad (8b)
\]
If \( u(t) \) is constant for \( T_0 k \leq t < T_0 k + T_s \) for constant \( T_s \) and \( k = 0, 1, 2, \ldots \), then \( x(T_s k) \) and \( y(T_s k) \) are exactly described by the discrete time state space system
\[
x(T_s k + T_s) = \Phi x(T_s k) + \Gamma u(T_s k) \quad (9a)
\]
\[
y(T_s k) = Cx(T_s k) + Du(T_s k), \quad (9b)
\]
where

\[ \Phi = e^{AT_s} \]  \hspace{2cm} (10)
\[ \Gamma = \int_0^{T_s} e^{A\tau}d\tau B. \]  \hspace{2cm} (11)

Proof. This is a well known result, and the derivation can be found in for example [3].

We will now examine how this result can be extended to descriptor systems. In the descriptor case, we cannot assume that the input is piecewise constant as the system according to Theorem 2.2 may depend on derivatives of the input. Instead we will assume that the \( m-1 \)th derivative (\( m \) is from the normal form in Theorem 2.2) is piecewise constant, as this is the highest derivative on which the system depends directly. We will assume that \( \det(sE - J) \neq 0 \) for all \( s \), so we can according to Theorem 2.2 assume that the system is on the form (7).

Our goal is to use lemma 2.1 to get a discrete representation. To do this we must first transform (7) to a form on which lemma 2.1 can be applied. To do this, we begin by defining a vector with the input and some of its derivatives:

\[ w_3(t) = \begin{pmatrix} u(t) \\ \dot{u}(t) \\ \vdots \\ u^{(m-2)}(t) \end{pmatrix} \]  \hspace{2cm} (12)

which leads to the relation

\[ \dot{w}_3(t) = \begin{pmatrix} \ddot{u}(t) \\ \dot{u}(t) \\ \vdots \\ u^{(m-1)}(t) \end{pmatrix} = \begin{pmatrix} 0 & I & \cdots & 0 \\ 0 & 0 & \cdots & I \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} w_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u^{(m-1)}(t) \]  \hspace{2cm} (13)

Now (7) can be rewritten to depend on \( w_3(t) \) instead of depending directly on the different derivatives of \( u(t) \). The new description of the solutions of (6) will be

\[ \dot{w}_1(t) = Aw_1(t) + \begin{pmatrix} B & 0 & \cdots & 0 \end{pmatrix} w_3(t) \]  \hspace{2cm} (14a)
\[ w_2(t) = \begin{pmatrix} I & (-N) & \cdots & (-N)^{m-2} \end{pmatrix} Dw_3(t) + (-N)^{m-1} Du^{(m-1)}(t) \]  \hspace{2cm} (14b)
\[ \dot{w}_3(t) = \begin{pmatrix} \ddot{u}(t) \\ \dot{u}(t) \\ \vdots \\ u^{(m-1)}(t) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} w_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} u^{(m-1)}(t) \]  \hspace{2cm} (14c)
\[ y(t) = LT^{-1} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} \]  \hspace{2cm} (14d)
If $w_2(t)$ is eliminated, these equations can be seen as a normal state space equation with $u^{(m-1)}(t)$ as the input:

$$\begin{pmatrix} \dot{w}_1(t) \\ \dot{w}_3(t) \end{pmatrix} = \begin{pmatrix} A & B & 0 & \ldots & 0 \\ 0 & 0 & I & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & I \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} u^{(m-1)}(t) (15a)$$

$$y(t) = LT^{-1} \begin{pmatrix} I & 0 & 0 & \ldots & 0 \\ 0 & D & (-N)D & \ldots & (-N)^{m-2}D \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_3(t) \end{pmatrix} +$$

$$+ LT^{-1} \begin{pmatrix} 0 \\ (-N)^{m-1}D \end{pmatrix} u^{(m-1)}(t) (15b)$$

If we let $x(t) = (w_1^T(t) \quad w_3^T(t) )^T$ this can be written in the compact form

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}u^{(m-1)}(t) \quad (16a)$$

$$y(t) = \tilde{C}x(t) + \tilde{D}u^{(m-1)}(t). \quad (16b)$$

Now, if we assume that $u^{(m-1)}(t)$ is piecewise constant, lemma 2.1 can be applied to (16) to give an exact discrete time description of the original time continuous descriptor system (1). We have thus arrived at the following theorem:

**Theorem 2.3.** Consider the time continuous descriptor system

$$E\dot{\xi}(t) = J\xi(t) + Ku(t) \quad (17a)$$

$$y(t) = L\xi(t) \quad (17b)$$

with the normal form

$$\dot{w}_1(t) = Aw_1(t) + Bu(t) \quad (18a)$$

$$w_2(t) = \sum_{i=0}^{m-1} (-N)^i Du^{(i)}(t) \quad (18b)$$

$$\xi(t) = T^{-1}w(t) \quad (18c)$$

$$y(t) = LT^{-1}w(t). \quad (18d)$$

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

$$x(T_sk + T_s) = \Phi x(T_sk) + \Gamma u^{(m-1)}(T_sk) \quad (19a)$$

$$y(T_sk) = \tilde{C}x(T_sk) + \tilde{D}u^{(m-1)}(T_s k). \quad (19b)$$
where

\[ \Phi = e^{\hat{A}T_s} \]
\[ \Gamma = \int_0^{T_s} e^{\hat{A} \tau} d\tau \hat{B}. \]

and \( \hat{A}, \hat{B}, \hat{C}, \) and \( \hat{D} \) are defined in (15).

Proof. The proof is provided by the discussion above.

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, for example that it is piecewise linear.

2.3 Noise model

Systems are often affected by other factors than the known inputs. These factors can from a modeling perspective be seen as unmeasured inputs. It is also possible that the measurement \( y(t) \) is corrupted by noise. If effects of the unmeasured inputs and the noise on the output is included in the model, we say that we have added a noise model.

In the case of time continuous descriptor systems we can add a noise model according to (22), where \( v_1(t) \) represents the unmodeled inputs and \( v_2(t) \) represents the measurement noise. \( M \) is a constant matrix.

\[
E_\dot{\xi}(t) = J\xi(t) + Ku(t) + M v_1(t) \\
y(t) = L\xi(t) + v_2(t)
\]

(22) can also be written as

\[
E_\dot{\xi}(t) = J\xi(t) + \begin{pmatrix} K & M \end{pmatrix} \begin{pmatrix} u(t) \\ v_1(t) \end{pmatrix} \\
y(t) = L\xi(t) + v_2(t).
\]

By applying Theorem 2.2 to (23) the solution can be described by

\[
\dot{w}_1(t) = Aw_1(t) + B_1u(t) + B_2v_1(t) \\
w_2(t) = \sum_{i=0}^{m-1} (-N)^i D_1u^{(i)}(t) + \sum_{i=0}^{m-1} (-N)^i D_2v_1^{(i)}(t) \\
\xi(t) = T^{-1}w(t) \\
y(t) = LT^{-1}w(t) + v_2(t).
\]

A normal assumption when \( E \) is the identity matrix (i.e. we have a normal state space description) is that \( v_1(t) \) and \( v_2(t) \) are white noise signals. Sometimes it is also assumed that they have a Gaussian distribution. The reason is
usually that the state estimation problem then is easy (the solution becomes the Kalman filter). To be able to estimate the states of (22) we would like to make the same assumption here. Time continuous white noise signals are however delicate mathematical objects, for example the integral and derivative are not well defined in the normal sense. See further [2]. Therefore one has for example to consider carefully what (22) means if \(v_1(t)\) and \(v_2(t)\) are assumed to be white noise signals. Furthermore, we cannot allow that any derivatives of \(v_1(t)\) occurs in (24), that is we have to require that

\[ ND_2 = 0. \]  

(25)

Note that (25) is related to the so called impulse controllability (see for example [8]) with respect to \(v_1(t)\), but this viewpoint is not further pursued here.

In [24] it is shown that (25) is equivalent to that the \(M\) matrix in (22) is in the range of a certain matrix.

Now, if we assume that the matrix \(M\) in (22) is such that (25) is fulfilled, the normal form (24) can be written as

\[
\begin{align*}
\dot{w}_1(t) &= Aw_1(t) + B_1 u(t) + B_2 v_1(t) \quad (26a) \\
\dot{w}_2(t) &= \sum_{i=0}^{m-1} (-N)^i D_1 u^{(i)}(t) + D_2 v_1(t) \quad (26b) \\
\xi(t) &= T^{-1} w(t) \quad (26c) \\
y(t) &= LT^{-1} w(t) + v_2(t). \quad (26d)
\end{align*}
\]

We now proceed to transform (26) to a state space description with \(u^{(m-1)}(t)\) as the input using the same method as in Section 2.1. We thus define \(w_3(t)\) according to (12), which puts us at the description

\[
\begin{align*}
\dot{w}_1(t) &= Aw_1(t) + \begin{pmatrix} B_1 & 0 & \ldots & 0 \end{pmatrix} w_3(t) + B_2 v_1(t) \quad (27a) \\
\dot{w}_2(t) &= \begin{pmatrix} I & (-N) & \ldots & (-N)^{m-2} \end{pmatrix} D_1 w_3(t) + \begin{pmatrix} 0 & \ldots & 0 \end{pmatrix} u^{(m-1)}(t) \\
&\quad + (-N)^{m-1} D_1 u^{(m-1)}(t) + D_2 v_1(t) \quad (27b) \\
\dot{w}_3(t) &= \begin{pmatrix} 0 & I & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & I \\ 0 & 0 & \ldots & 0 \end{pmatrix} w_3(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ v_2(t) \end{pmatrix} \quad (27c) \\
y(t) &= LT^{-1} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} + v_2(t). \quad (27d)
\end{align*}
\]

Eliminating \(w_2(t)\) and stacking \(w_1(t)\) and \(w_3(t)\) together now gives us the de-
Deﬁning $x(t) = \begin{bmatrix} w_1(t) \\ w_3(t) \end{bmatrix}$ now gives us the more compact notation

$$\begin{align*}
\dot{x}(t) &= \tilde{A}x(t) + \tilde{B}u^{(m-1)}(t) + \tilde{B}_2v_1(t) \\
y(t) &= \tilde{C}x(t) + \tilde{D}u^{(m-1)}(t) + (\tilde{N} \ I ) \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}.
\end{align*}$$

(29a)

(29b)

We have thus shown that it is possible to write down a state space system with a noise model which has the same solution as the descriptor system with noise model (22) under the assumption (25) that $ND_2 = 0$. Note however, that in the state space model, the noise on the output equation is related to the noise on the state equation through the $v_1(t)$ term. This relation can however be eliminated if we make the stronger assumption that $D_2 = 0$. If $D_2 = 0$, then $\tilde{N} = 0$ so the state space description simpliﬁes to

$$\begin{align*}
\dot{x}(t) &= \tilde{A}x(t) + \tilde{B}u^{(m-1)}(t) + \tilde{B}_2v_1(t) \\
y(t) &= \tilde{C}x(t) + \tilde{D}u^{(m-1)}(t) + v_2(t).
\end{align*}$$

(30a)

(30b)

Here, the noise on the state and output equations are related only if $v_1(t)$ and $v_2(t)$ are.
2.3.1 Discretization

Also when we have a noise model, it is interesting to examine what the discrete time description of the time continuous descriptor system is. An assumption which makes it possible to utilize the results in Section 2.1 is that the noise is piecewise constant. It might be argued that it is not physical to assume that the noise is piecewise constant, but if the sampling time is small enough, it might be a reasonable assumption. (Note that we have not yet assumed that the noise is white!) Anyway, making this assumption leads to the following theorem:

**Theorem 2.4.** Consider a time continuous descriptor system with a noise model

\[ E\dot{\xi}(t) = J\xi(t) + Ku(t) + Mv_1(t) \]  
\[ y(t) = L\xi(t) + v_2(t) \]

with the normal form

\[ \dot{w}_1(t) = Aw_1(t) + B_1u(t) + B_2v_1(t) \]  
\[ w_2(t) = \sum_{i=0}^{m-1} (-N)^i D_1 u^{(i)}(t) + D_2 v_1(t) \]  
\[ \xi(t) = T^{-1} w(t) \]  
\[ y(t) = LT^{-1} w(t) + v_2(t). \]

(Note that not all \( M \) matrices may allow such a normal form.)

If \( u^{(m-1)}(t) \) and \( v_1(t) \) are constant for \( T_s k \leq t < T_s k + T_s \) for constant \( T_s \) and \( k = 0, 1, 2, \ldots \), then \( y(T_s k) \) is exactly described by the discrete time state space system

\[ x(T_s k + T_s) = \Phi x(T_s k) + \Gamma_1 u^{(m-1)}(T_s k) + \Gamma_2 v_1(T_s k) \]
\[ y(T_s k) = \tilde{C} x(T_s k) + \tilde{D} u^{(m-1)}(T_s k) + \left( \begin{array}{c} N \ I \end{array} \right) \left( \begin{array}{c} v_1(t) \ v_2(t) \end{array} \right). \]

where

\[ \Phi = e^{\tilde{A} T_s} \]  
\[ \Gamma_1 = \int_0^{T_s} e^{\tilde{A} \tau} d\tau \tilde{B} \]  
\[ \Gamma_2 = \int_0^{T_s} e^{\tilde{A} \tau} d\tau \tilde{B}_2. \]

and \( \tilde{A}, \tilde{B}, \tilde{B}_2, \tilde{C}, \) and \( \tilde{D} \) are defined in (28).

Note that the assumption that \( \tilde{N} = 0 \) which was discussed earlier simplifies the expression (33).
Proof. Rewriting (29) as

\begin{align*}
\dot{x}(t) &= \tilde{A}x(t) + \begin{pmatrix} \tilde{B} \\
\tilde{B}_2 & B \end{pmatrix} \begin{pmatrix} u^{(m-1)}(t) \\
v_1(t) \end{pmatrix} \\
y(t) &= \tilde{C}x(t) + \tilde{D}u^{(m-1)}(t) + \begin{pmatrix} \tilde{N} \\
I \end{pmatrix} \begin{pmatrix} v_1(t) \\
v_2(t) \end{pmatrix}
\end{align*}

(37a) (37b)

allows us to apply lemma 2.1 and we are done.

The assumption which led to Theorem 2.4 was that the noise was piecewise constant. If we do not want to make this assumption, we can instead make the assumption that \(v_1(t)\) is white noise in (29) and (30). In this case, formulas for transforming the state space noise in (29) and (30) to discrete time are available in for example [2] and are not repeated here.

2.3.2 A shortcut

We went through a lot of work to examine what the corresponding state space description and discrete time description of the time continuous descriptor system with noise model (22) was. If we do not have any prior knowledge about the noise model, and want to identify it as a black box model, it might be better to add the noise model after we have converted the descriptor system (without noise model) to the state space form (16), or even after we have discretized the system. Then we do not have to make the conversion of the noise model, and we do not have to consider the assumption (25).

2.4 Kalman filtering

We have now established that it under some assumptions is possible to transfer a continuous time descriptor system to 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 (38). The implementation of a Kalman filter is then straightforward, see e.g., [1].

\begin{align*}
x(t+1) &= Ax(t) + Bu(t) + Nv_1(t) \\
y(t) &= Cx(t) + Du(t) + v_2(t)
\end{align*}

(38a) (38b)

Note that the discretized descriptor system (33) is just a special case of this model with proper time scaling and redefinitions. To transform (33) into (38) we need to do the following:

- Re-scale the time indices with the factor \(T_s\).
- Rename matrices according to \(A \rightarrow A, \Gamma_1 \rightarrow B, \Gamma_2 \rightarrow N, \tilde{C} \rightarrow C, \) and \(\tilde{D} \rightarrow D\).
- Rename the input signal, \(u^{(m-1)}(t) \rightarrow u(t)\).
 Rename the output noise sequence according to \( (\begin{bmatrix} \hat{N} & I \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}) \rightarrow v_2(t) \). Note that this leads to changed correlation matrices for and cross correlation between the new \( v_1(t) \) and \( v_2(t) \).

Note that we through a normal Kalman filter only get estimations of the state vector \( x(t) \) and the output \( y(t) \), not of the vector \( \xi(t) \). But if we want estimations of the full vector \( \xi(t) \), all we have to do is to select \( L \) in (31) as the identity matrix so that \( y(t) = \xi(t) \).

### 2.5 Time domain identification

As it has been shown that the time continuous DAE system under some assumptions can be described by a discrete time state space system, a large amount of theory for system identification is available, e.g., [16]. The probably most common method available is the prediction error method with a quadratic criterion. This method is based on using the Kalman filter as a one-step ahead predictor and then minimizing the prediction error. It is also possible to do maximum likelihood estimation if \( v_1(t) \) and \( v_2(t) \) are assumed to have a Gaussian distribution. Then the Kalman filter is used to calculate the probability distribution of the output, and the parameters are estimated by maximizing the probability of the measured output.

### 2.6 Frequency domain identification

The work which has been done until now has been based on transforming the descriptor system to the so called normal form. The normal form always exists if \( \det(sE - J) \) is not zero for all \( s \), according to the appendix. It is however not fully investigated if it always is practically possible to calculate the normal form, for example one might encounter numerical difficulties. The state space form which the normal form produces might also not always be well suited for estimation purposes, especially as we might have to differentiate the input numerically.

A way around these problems turns out to be identification in the frequency domain in continuous time, were it is assumed that the model structure is specified by transfer functions (or matrices of transfer functions) according to (39).

\[
Y(\omega) = G(i\omega, \theta)U(\omega) + H(i\omega, \theta)V_1(\omega) + V_2(\omega) \tag{39}
\]

Here \( \theta \) represents the parameters which are to be estimated. \( Y(\omega) \) and \( U(\omega) \) are the Fourier transforms of \( y(t) \) and \( u(t) \) respectively and \( G(i\omega, \theta) \) and \( H(i\omega, \theta) \) are transfer functions. \( V_1(\omega) \) and \( V_2(\omega) \) are the spectrums of the noise signals \( v_1(t) \) and \( v_2(t) \).

It turns out that a linear time continuous descriptor system with a noise model

\[
\begin{align*}
E \dot{\xi}(t) &= J\xi(t) + Ku(t) + Mv_1(t) \\
y(t) &= L\xi(t) + v_2(t)
\end{align*} \tag{40a-40b}
\]

Here \( E \) is the descriptor matrix, \( J \) is the index matrix, \( K \) is the input matrix, \( M \) is the noise input matrix, \( L \) is the output matrix, and \( \xi(t) \) is the state vector.
where \( v_1(t) \) and \( v_2(t) \) are assumed to be white noise signals can be transformed to the form (39) under the usual assumption that \( \text{det}(sE - J) \) is not zero for all \( s \). The only difference from the transfer function of a normal state space system is that \( G \) and \( H \) may not be proper in the general case. As \( v_1(t) \) is assumed to be white noise it is however assumed that the \( M \) matrix is such that \( H \) is proper.

To show how (40) can be transformed to the frequency domain we begin by Fourier transforming the equations to get (here, \( v_1(t) \) and \( v_2(t) \) are assumed to be transformable signals)

\[
\begin{align*}
Ei\omega \zeta(\omega) &= J\zeta(\omega) + KU(\omega) + MV_1(\omega) \\
Y(\omega) &= LE(\omega) + V_2(\omega)
\end{align*}
\] (41a) (41b)

We then get the desired description (42) by eliminating \( \zeta(\omega) \) (which can be done when \( (sE - J) \) is invertible).

\[
Y(\omega) = \underbrace{L(i\omega E - J)^{-1}KU(\omega)}_{G(i\omega)} + \underbrace{L(i\omega E - J)^{-1}MV_1(\omega)}_{H(i\omega)} + V_2(\omega)
\] (42)

Now that we have established that a descriptor system can be described by a transfer function, all we have to do is to plug the transfer function into any identification algorithm for the frequency domain. Books which treat this are [16, 22]. One possible selection of identification method is to minimize the criterion

\[
V(\theta) = \sum_{k=1}^{N} \left| Y(\omega_k) - G(i\omega_k, \theta)U(\omega_k) \right|^2 \cdot \frac{1}{|H(i\omega_k, \theta)|^2}
\] (43)

with respect to the parameters \( \theta \).

Note that we need the Fourier transforms of the input and output signals, but these can be estimated from time domain data.

### 3 An example

In this section we exemplify the algorithms examined in Section 2 by identifying parameters in a physical system. The system setup examined is a model of a generator, see Figure 1.

The model can be described as follows: The input to the model is the angle \( \theta_{in} \) on the left axis. This axis is connected to a rotating mass with inertia \( J \) which is rotated the angle \( \theta \) and rotates with the angular velocity \( \omega \). The torque acting on the left side of the mass is \( M_1 \) and the torque on the right side is \( M_2 \). The mass is then connected to second axis which is connected to the actual generator. The variables describing the second axis and the electrical quantities are then assumed to depend on each other in the following way:

\[
\begin{align*}
M_2 &= kI \\
u_1 &= k\omega
\end{align*}
\] (44a) (44b)
for some constant \( k \). The rest of the electrical circuit consists of two resistors and one inductor. The measured output is the voltage \( u_4 \).

Note that it here is easy to spot that too many variables are included in the model if the torques not are interesting, as the mass then will have no impact on the system. This is to show that the identification algorithm treats this automatically. (In larger models it may not be more difficult to spot that some variables are unnecessary.)

To generate input and output data for identification experiments, the model was simulated in the MathModelica add-on [12] for Mathematica [25]. Data was collected during 20 s with the sampling interval 0.01 s. The input \( \theta_{in} \) was selected as a chirp signal. The component values were:

\[
\begin{align*}
J &= 1 \text{ kg m}^2 \\
k &= -1 \text{ N m/A} \\
L &= 0.1 \text{ H} \\
R_1 &= 1 \Omega \\
R_2 &= 1 \Omega 
\end{align*}
\]

We now model the system in an object-oriented manner by just writing down the equations describing the different parts and the ways they are connected (these equations could have been generated in an automated way):

\[
\begin{align*}
\theta_{in} &= \theta \\
\dot{\theta} &= \omega \\
J \ddot{\omega}_1 &= M_1 + M_2 \\
M_2 &= k I \\
I_1 &= k \omega \\
u_2 &= L \dot{I} \\
u_3 &= R_1 I \\
u_4 &= R_2 I \\
u_1 &= u_2 + u_3 + u_4
\end{align*}
\]
To apply the identification methods examined in Section 2, we need to write these equations on the form (1). This is just a matter of simple rearrangement of the equations, and the result is

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & J & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & L & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
M_1 \\
M_2 \\
\dot{M}_1 \\
\dot{M}_2 \\
\ddot{\omega} \\
\dot{\theta} \\
\dot{i} \\
\dot{u}_1 \\
\dot{u}_2 \\
\dot{u}_3 \\
\dot{u}_4 \\
\end{pmatrix}
+ \begin{pmatrix}
\begin{pmatrix}
M_1 \\
M_2 \\
\omega \\
\theta \\
i \\
\dot{u}_1 \\
\dot{u}_2 \\
\dot{u}_3 \\
\dot{u}_4 \\
\end{pmatrix} \\
\end{pmatrix}
\begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
\theta_{\text{in}} \\
\end{pmatrix}
\]

(47a)

\[
y = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

(47b)

if the voltage \(u_4\) is considered to be the output. This model is now in such a form so that we can apply the algorithms discussed earlier. During the identification process both time and frequency domain methods will be used. The values \(R_1\) and \(R_2\) are assumed to be known (measured beforehand), so the values of \(L\) and \(k\) are to be estimated. (The value of \(J\) will not affect the output, see further below). Below the unknown parameters are identified both with time and frequency domain methods.

### 3.1 Time domain identification

The first step when identifying in the time domain is to transform the model to the form (16) (here, no noise model is added before the transformation). The
The transformed system is

\[
\dot{x} = -\frac{R_1 + R_2}{L} x + \frac{k^2(R_1 + R_2)}{L^2} \theta_{in} \tag{48a}
\]

\[
y = -\frac{R_1}{k} x + \frac{R_1 k}{L} \theta_{in} \tag{48b}
\]

The system actually has only one state, something which not was obvious from the original equations. The actual identification of the parameters is now performed with the System identification toolbox for MATLAB [17], that is the discretization and optimization with respect to the parameters is treated by the toolbox. The minimization is performed with the prediction error method with quadratic criterion. The optimization is performed both with an output error model (i.e., \(B_2\) in (30) is assumed to be zero) and by estimating \(B_2\) in (30) along with the parameters (i.e., with a black box noise model). The results are summarized in Table 1. As can be seen in the table, an output error model here gives more accurate estimates, but is more sensitive to the starting values.

### Table 1: Time domain identification.

<table>
<thead>
<tr>
<th>Starting values</th>
<th>Noise model</th>
<th>Estimated values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L = 10)H</td>
<td>Output error</td>
<td>(L = 0.12)H</td>
</tr>
<tr>
<td>(k = -10)Nm/A</td>
<td></td>
<td>(k = -1.0)Nm/A</td>
</tr>
<tr>
<td>(L = 1)H</td>
<td>Output error</td>
<td>Divergence</td>
</tr>
<tr>
<td>(k = 1)Nm/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L = 10)H</td>
<td>Black box</td>
<td>(L = 0.13)H</td>
</tr>
<tr>
<td>(k = -10)Nm/A</td>
<td></td>
<td>(k = -1.1)Nm/A</td>
</tr>
<tr>
<td>(L = 1)H</td>
<td>Black box</td>
<td>(L = 0.13)H</td>
</tr>
<tr>
<td>(k = 1)Nm/A</td>
<td></td>
<td>(k = -1.2)Nm/A</td>
</tr>
</tbody>
</table>

#### 3.2 Frequency domain identification

To do identification in the frequency domain, the transfer function for the model must be calculated. This can be done using equation (42) (it is assumed that \(M = 0\) in this equation, i.e., an output error model is selected). The result is

\[
G(s) = \frac{R_2 k s}{Ls + R_1 + R_2}. \tag{49}
\]

The necessary frequency domain data is produced by simply taking the fast Fourier transform (FFT) of the time domain data. The identification is then performed by minimizing the criterion

\[
\sum_{\omega_i} \left| G(i\omega_i) - \hat{G}(i\omega_i) \right|^2 \tag{50}
\]

with respect to \(k\) and \(L\). \(\hat{G}(i\omega_i)\) is the transfer function which was estimated by transforming the time domain data, and \(\omega_i\) are the frequencies at which this transfer function is estimated. The result is summarized in Table 2.
Table 2: Frequency domain identification.

<table>
<thead>
<tr>
<th>Starting values</th>
<th>Estimated values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 10, \text{H}$</td>
<td>$L = 0.12, \text{H}$</td>
</tr>
<tr>
<td>$k = -10, \text{Nm/A}$</td>
<td>$k = -1.1, \text{Nm/A}$</td>
</tr>
<tr>
<td>$L = 1, \text{H}$</td>
<td>$L = 0.12, \text{H}$</td>
</tr>
<tr>
<td>$k = 1, \text{Nm/A}$</td>
<td>$k = -1.1, \text{Nm/A}$</td>
</tr>
</tbody>
</table>

As can be seen in the table, the results when identifying in the frequency domain are comparable to the ones when identifying in the time domain.

4 Time discrete descriptor systems

The problem of identifying parameters in the discrete time descriptor system (2) is closely related to the continuous time problem in the sense that the theory is similar. Therefore, the results in this section will mainly be a description of how the results for continuous time descriptor systems carry over to the discrete time case.

4.1 Basic facts

In this section two basic facts about discrete linear descriptor systems are established, namely the solvability of (2) and the possibility to put (2) on a normal form. These results are similar to the time continuous case.

4.1.1 Solvability

As in the time continuous case, the solvability of (2) can be said to be that there should exist a unique solution $\xi(t)$. The solvability turns out to be dependent only on the matrices $E$ and $J$. We formalize the solvability with a definition.

**Definition 4.1 (Solvability).** Solvability of the discrete linear descriptor system

$$E\xi(t + 1) - J\xi(t) = Ku(t) \tag{51}$$

is the existence of a unique solution $\xi(t)$, $t \geq 0$, for every input sequence $u(t)$ with $u(0) = 0$ and $\xi(0) = 0$.

If we restrict ourselves to sequences $\xi(t)$ and $u(t)$ that can be $Z$-transformed we get the following criterion for checking if a time discrete descriptor system is solvable.

**Theorem 4.1.** If only sequences $\xi(t)$ and $u(t)$ that can be $Z$-transformed are considered, solvability of (51) is equivalent to that $\det(zE - J)$ not is zero for all $z$. 

16
Proof. First transform (51) to get \((\xi(0) = 0)\)
\[
(zE - J)Z[\xi(t)] = KZ[u(t)].
\] (52)

Now assume that \(\det(zE - J)\) is not zero for all \(z\). This implies that the inverse exists, and we proceed to multiply (52) from the left with \((zE - J)^{-1}\) and get
\[
Z[\xi(t)] = (zE - J)^{-1}KZ[u(t)].
\] (53)

The sequence \(\xi(t)\) is thus uniquely determined as the inverse \(Z\)-transform of the right hand side of (53).

On the contrary, assume that \(\det(zE - J)\) is zero for all \(z\). We can then choose a vector \(\alpha(z) \neq 0\) such that \((zE - J)\alpha(z) = 0\). This gives that if \(Z[\xi(t)]\) is a solution of (52), then so is \(Z[\xi(t)] + \alpha(z)\) so the solution is not unique. \(\square\)

Also note that if \(\det(zE - J)\) is zero for all \(z\), a solution \(\xi(t)\) which satisfies (51) may not even exist. The definition of solvability which is presented here is not the same as in for example [18, 19]. Both definitions do however lead to Theorem 4.1.

To our knowledge, solvability is a reasonable requirement to put on a model of a physical systems, and therefore only solvable systems are considered in this report.

4.1.2 The normal form

In this section we examine what the solution of (2) is, assuming that \(zE - J\) is invertible. The result is similar to the ones for continuous time systems in Section 2.1.2, the only differences are that the derivatives are replaced with time shifts.

**Theorem 4.2.** If \(\det(zE - J)\) not is zero for all \(z\), the solution of
\[
E\xi(t + 1) = J\xi(t) + Ku(t)
\]
\[
y(t) = L\xi(t)
\] (54a,b)
can be described by
\[
w_1(t + 1) = Aw_1(t) + Bu(t)
\] (55a)
\[
w_2(t) = \sum_{i=0}^{m-1} (-N)^i Du(t + i)
\] (55b)
\[
\xi(t) = T^{-1}w(t)
\] (55c)
\[
y(t) = LT^{-1}w(t)
\] (55d)

with notation from appendix A.

Proof. The result follows directly from corollary A.1 in the appendix. \(\square\)

Note that the system is non-causal in the general case as the output depends on future values of the input.
4.2 Noise model

Similarly to how a noise model is added in the time continuous case, a noise model can be added to discrete time descriptor systems according to (56). $v_1(t)$ and $v_2(t)$ are uncorrelated sequences of white noise and $M$ is a constant matrix.

\[
E\xi(t+1) = J\xi(t) + Ku(t) + Mv_1(t) \quad (56a)
\]

\[
y(t) = L\xi(t) + v_2(t) \quad (56b)
\]

The noisy system might be non-causal not only with respect to the input signal, but also with respect to the noise. Actually, the solutions can be described by (57). This can be realized by stacking $u(t)$ and $v_1(t)$ in one vector and then applying Theorem 4.2.

\[
w_1(t+1) = Aw_1(t) + B_1u(t) + B_2v_1(t) \quad (57a)
\]

\[
w_2(t) = \sum_{i=0}^{m-1} (-N)^iD_1u(t+i) + \sum_{i=0}^{m-1} (-N)^iD_2v_1(t+i) \quad (57b)
\]

\[
\xi(t) = T^{-1}w(t) \quad (57c)
\]

\[
y(t) = LT^{-1}w(t) + v_2(t) \quad (57d)
\]

A difference from the time continuous 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. As is pointed out in [24] the dependence on future noise could be handled by time shifting the noise sequence. It can however be argued that dependence on future noise values is not physical, so another approach may be to use the assumption (25).

4.3 Kalman filtering

As in the time continuous case, the normal form can be rewritten to a state space form which then makes it possible to implement a Kalman filter to estimate states and make predictions. One thing which makes matters easier in the discrete time case is that the state space system does not have to be discretized. The derivation of the state space form is similar to the time continuous case, so it is omitted. Here we however have to consider $u(t+m-1)$ instead of $u^{(m-1)}(t)$ as the input. If no noise model is included, the state space form is

\[
x(t+1) = \tilde{Ax}(t) + \tilde{Bu}(t+m-1) \quad (58a)
\]

\[
y(t) = \tilde{Cx}(t) + \tilde{Du}(t+m-1) \quad (58b)
\]

with definitions from (15). If a noise model is included and we assume that (25) holds, the state space form is

\[
\dot{x}(t) = \tilde{Ax}(t) + \tilde{Bu}^{(m-1)}(t) + \tilde{B}_2v_1(t) \quad (59a)
\]

\[
y(t) = \tilde{Cx}(t) + \tilde{Du}^{(m-1)}(t) + \left( \tilde{N} \quad I \right) \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} \quad (59b)
\]
with definitions from (28). When the discrete time descriptor system has been converted to state space form implementation of the Kalman filter is straightforward, see e.g., [1].

Previous work on kalman filtering of discrete descriptor systems is among others [10, 20, 21, 9, 6, 7]. The approach taken here is similar to the one in [6].

### 4.4 Time domain identification

As the time discrete descriptor systems can be transformed into a time discrete state space system, the same methods as are available in the time continuous case are available to us here. That is we can for example use a prediction error method or maximum likelihood estimation.

### 4.5 Frequency domain identification

Also identification in the frequency domain is similar to the time continuous case. If (56) is transformed to the frequency domain we get

\[
Y(\omega) = G(e^{i\omega})U(\omega) + H(e^{i\omega})V_1(\omega) + V_2(\omega)
\]

where

\[
G(z) = L(zE - J)^{-1}K
\]

and

\[
H(z) = L(zE - J)^{-1}M.
\]

This transformation can always be made if \(zE - J\) is invertible. The only difference from the transfer functions of a normal state space system is that \(G(z)\) and \(H(z)\) might not be causal transfer functions. This can be realized by looking at the normal form, which shows that the descriptor system may not be causal. Therefore, standard methods for identification in the frequency domain can be used, see [16, 22].

### 5 Conclusions

This report has treated how the parameter identification problem for descriptor systems can be solved for the time invariant linear case, both in discrete and continuous time. In the time domain, maximum likelihood identification can be implemented by constructing a Kalman filter. This requires the normal form of the linear descriptor system. Although the normal form can always be calculated in theory, it is a matter for future research on how it can be computed in a numerically stable way. Therefore it might be better to do identification directly in the frequency domain in some cases, as we then do not need to calculate to normal form.

The report treats only linear systems, but the result may also have some application to nonlinear descriptor systems through linearization. Linearization of descriptor systems is treated in [5].
A Proof of the normal form

This section is partly taken from [14]. The proofs of the normal forms are carried out for the continuous time case. They do however also hold for the discrete time case if all derivatives are changed to time shifts, for example $\dot{x}(t) \rightarrow x(t + 1)$.

**Definition A.1 (Nilpotent matrix).** A nilpotent matrix of order $k$ is a matrix $N$ such that $N^k = 0$ and $N^{k-1} \neq 0$.

**Lemma A.1.** For every matrix $A$, there exists a matrix $T$ such that

$$TAT^{-1} = \begin{pmatrix} M & 0 \\ 0 & N \end{pmatrix}$$

where $M$ is non-singular and $N$ is nilpotent.

**Proof.** This is a consequence of Fitting’s lemma, see [15].

Fitting’s lemma also gives that $T$ can be calculated in the following way: Let $k$ be the smallest integer such that $N(A^k) = N(A^{k+1})$ and let $v_1, ..., v_r$ span $N(A^k)$ and $f_1, ..., f_s$ span $V(A^k)$. $N(X)$ is the null space of $X$ and $V(X)$ is the range of $X$. Then we can choose $T$ as

$$T^{-1} = [f_1 \ldots f_s v_1 \ldots v_r].$$

**Theorem A.1 (The normal form).** Consider the linear descriptor system

$$E \dot{\xi}(t) + F \xi(t) = Gu(t).$$

If $\det(\lambda E + F)$ not is zero for all $\lambda \in \mathbb{R}$, then there exist non-singular matrices $P$ and $Q$ such that the transformation

$$PEQ \dot{\tilde{w}}(t) + PFQ \tilde{w}(t) = PGu(t)$$

$$\xi(t) = Q\tilde{w}(t)$$

gives the model

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{\tilde{w}}_1(t) \\ \dot{\tilde{w}}_2(t) \end{pmatrix} + \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} = \begin{pmatrix} B \\ D \end{pmatrix} u(t)$$

where $N$ is nilpotent.

**Proof.** Rewrite (65) as

$$E(\dot{\xi}(t) - \lambda \xi(t)) + (\lambda E + F)\xi(t) = Gu(t)$$

and multiply from the left with $(\lambda E + F)^{-1}$. $\lambda$ is a scalar that is chosen so that the inverse exists. This is possible since the system is solvable. This gives

$$\underbrace{(\lambda E + F)^{-1}E(\dot{\xi}(t) - \lambda \xi(t))}_E + \underbrace{\xi(t)}_{\tilde{\xi}} = \underbrace{(\lambda E + F)^{-1}Gu(t)}_{\tilde{g}}.$$

21
Now, according to lemma A.1 we can find a $T$ which transforms $\dot{E}$ according to (63). Making the variable transformation

$$\xi(t) = T^{-1}w(t)$$

and multiplying (70) from the left with $T$ gives

$$T\dot{E}T^{-1}[\dot{w}(t) - \lambda w(t)] + w(t) = T\dot{G}u(t).$$

Now, as $T$ is selected according to lemma A.1, this can be written as

$$M\dot{w}_1(t) + \left[ I - \lambda M \begin{pmatrix} 0 & 0 \\ 0 & I - \lambda N \end{pmatrix} \right] \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} = T\dot{G}u(t)$$

where $M$ is non-singular and $\tilde{N}$ is nilpotent. Multiplication from the left with

$$(I - \lambda \tilde{N})^{-1}$$

exists as $\tilde{N}$ is nilpotent) now finally puts us at the desired form.

The proof gives that the matrices $P$ and $Q$ can be calculated as

$$P = \begin{pmatrix} M^{-1} & 0 \\ 0 & (I - \lambda \tilde{N})^{-1} \end{pmatrix} T(\lambda E + F)^{-1}$$

$$Q = T^{-1}.$$  

\[ \square \]

**Corollary A.1 (Explicit solution).** If $\det(\lambda E + F)$ not is zero for all $\lambda \in \mathbb{R}$, the linear descriptor system

$$E\ddot{\xi}(t) + F\dot{\xi}(t) = Gu(t).$$

has the solution

$$\dot{w}_1(t) = Aw_1(t) + Bu(t)$$

$$w_2(t) = \sum_{i=0}^{m-1} (-N)^i Du^{(i)}(t)$$

$$\xi(t) = T^{-1}w(t)$$

with notation from Theorem A.1.

**Proof.** According to Theorem A.1 we can without loss of generality assume that the system is on the form

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{w}_1(t) \\ \dot{w}_2(t) \end{pmatrix} + \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} = \begin{pmatrix} B \\ D \end{pmatrix} u(t)$$

$$\xi(t) = T^{-1} \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix}.$$
Now, if \( N = 0 \) we have that
\[
  w_2(t) = Du(t)
\]  
(80)
and we are done. If \( N \neq 0 \) we can multiply the second row of (79a) with \( N \) and get
\[
  N^2 \hat{w}_2(t) + Nw(t) = NDu(t).
\]  
(81)
We now differentiate (81) and insert the second row of (79a). This gives
\[
  w_2(t) = Du(t) - ND\hat{u}(t) + N^2 \hat{w}_2(t)
\]  
(82)
If \( N^2 = 0 \) we are done, otherwise we just continue until \( N^m = 0 \) (this is true for some \( m \) as \( N \) is nilpotent). We would then arrive at an expression like
\[
  w_2(t) = \sum_{i=0}^{m-1} (-N)^i Du^{(i)}(t)
\]  
(83)
and we are done.
References


