Robustness Analysis of Large Differential-Algebraic Systems with Parametric Uncertainty

Michael Lantz, Anders Rantzer
Department of Automatic Control
Lund Institute of Technology, Box 118, Lund, Sweden
Fax +46-46-138118, Email: rantzer@control.lth.se

Abstract
A general computational methodology is introduced for analysis of parametric uncertainty in large differential-algebraic systems. The objective is to convert the analysis problem into computation of structured singular values, while at the same time keeping the matrix dimensions down.

The proposed methodology has been implemented using the Omsim package for object-oriented modeling and Matlab for matrix computations. It is applied to a model of the Scandinavian power system, including 16 generators, 16 power loads and 20 transmission lines. There are totally 16 inputs, 16 outputs, 63 states and more than 500 parameters, of which 44 are assumed to be uncertain.

1. Introduction
Methods for robustness analysis with respect to parametric uncertainty has during the last decade been developed based on computations of the structured singular value $\mu$. It has been verified that every rational transfer matrix with rational dependence on some uncertain parameters can in principle be analysed with respect to stability and $L_2$-gain performance using such computations.

However, even if the algorithms for structured singular value computations can handle matrices of dimension as high as 50-100, not many applications of this size have been published. One of the reasons is that proper generation of input data for large problems is a non-trivial task. A typical scenario would be the following: Using a large and complicated differential-algebraic model, a system has been simulated and a stable equilibrium has been found. The objective is to compute the maximal range of parameter variations for which this equilibrium remains locally stable. To do this, one has to linearize the system around the equilibrium point and then rewrite the resulting parameter dependent transfer matrix as a feedback loop involving a constant transfer matrix $M(s)$ and a diagonal parameter matrix $\Delta = \text{diag}\{\delta_1, \ldots, \delta_m\}$.

Our focus is on the transition from the linearization to the transfer matrix $M(s)$. This problem is far from trivial and earlier publications [14], [10], [15], leave room for many improvements. In this paper, it is assumed that the linearization consists
of a number of algebraic and differential equations with uncertain physical parameters appearing linearly in the coefficients. This is natural for many physical models and gives the same generality as state space descriptions with rational parameter dependence in the state matrix.

As an application of the methodology, we analyse a model for the Scandinavian power system network. The model is built componentwise in the object oriented modeling tool OmSim, [2], [3], [4] and the software generates Matlab m-files with the system matrices on symbolic form. A Matlab script then does the conversion to the input format of the µ-toolbox along the lines described below.

2. Preliminaries

This section will present a short review of the structured singular value µ and the concepts associated with it. The short presentation is based on [15] and [7].

Definition of µ

The µ has been an important issue of robustness analysis since it was proposed in the beginning of the 80s, as it makes it possible to analyze structured uncertainties. For example, parametric uncertainties is ideal to analyze using the µ concept, which is why it is considered here.

Definition 1

The µ-value or the Structured Singular Value is defined by

\[ \mu_\Delta(M) = \frac{1}{\min\{\sigma(\Delta) : \det(I - M\Delta) = 0\}} \]

unless no \( \Delta \) makes \( I - M\Delta \) singular, in which case \( \mu_\Delta = 0 \).

In words, the µ-value is the inverse of the smallest perturbation that causes \( I - M\Delta \) to become singular. Note that the µ-value usually is frequency dependent since \( M \) usually is frequency dependent.

It is difficult to calculate the µ exactly when the dimensions and structure of \( \Delta \) and \( M \) rise, but it has been discussed in several papers the prospect of finding good and easily computable upper and lower bounds for the µ. Such discussions can be found in e.g [7], [8] and [15].

Robust stability

The µ-value as defined in definition 1 can be used to determine whether a specified dynamic system fulfills robust stability, for a certain set of uncertainties. In order to use µ for this kind of robustness analysis, we need to write our system on a certain form with the uncertainties extracted from the model and in a feedback loop. Consider Figure 1.

Consider a block partitioning of the matrix \( M \) as

\[ M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2) \times (q_1+q_2)}, \]

In order to analyze the robust stability of the system, we examine the \((M_{11}, \Delta)\)-loop and we will obtain robust stability as described in Theorem 1.
Figure 1  Linear system with uncertainty feedback matrix.

**Theorem 1**
The system described by Figure 1 is robustly stable for all $\Delta$ such that $\bar{\sigma}(\Delta) < 1$ if and only if
\[
\mu(\Delta) < 1, \quad \forall \omega \in [0, \infty].
\]

This gives a good criterion for determining robust stability. In engineering applications, however, the robust stability criterion is often not directly applied, as the engineer is more interested in the input-output behavior of the model, which yields the performance of the model. If the performance is bad, then robust stability is not a good enough condition. We will consider this in the next section.

**Robust performance**
Let us now examine the full model as it was given in Figure 1. It is easy to imagine that the concept of robust stability discussed in the previous section may not be enough for analysis of a system as we ignored the influence of the inputs on the system and only considered internal stability.

Robust performance can be examined by closing the loop corresponding to the inputs and outputs with another feedback $\Delta_F$. The resulting system can be seen in Figure 2.

Figure 2  Closed loop for analyzing robust performance.

The fictitious uncertainty block, $\Delta_F$, is a full complex uncertainty block. This makes the problem virtually the same as the one we had when we considered robust stability. The $\mu$-values can be calculated in the same way as before, with the difference that we now have another augmented perturbation matrix, $\Delta_P$, given by

\[
\Delta_P = \begin{bmatrix}
\Delta & 0 \\
0 & \Delta_F
\end{bmatrix}.
\]
We naturally also use the complete transfer function matrix $M$ during the analysis, and thus we get the different $\mu$-values for the robust performance test.

In the same way as before we draw our conclusions of the robustness from these $\mu$-values. The well known Theorem 2 is often used.

**Theorem 2**
The following expressions are equivalent:

$$
\mu_{\Delta P}(M) < 1 \iff \begin{cases} 
\mu_{\Delta}(M_{11}) < 1 \\
\| M_{22} + M_{21} \Delta (I - M_{11} \Delta) M_{12} \| < 1 
\end{cases}
$$

The equivalence given in Theorem 2 implies that if the $\mu$-value given from the system in Figure 2 is less than 1, we can draw the conclusions that the system given by Figure 1 is robustly stable and that it fulfills the performance criterion that $\| y \|_2 < \| u \|_2$. The system is then said to perform robustly.

### 3. Object-oriented Modeling in Omola

In this section we discuss the concept of object-oriented modeling in the modeling language Omola, and how we can use this modeling tool to provide a set of model equations which in the next section will be used in order to provide an description with uncertainty feedback as we have seen in Figure 1.

Only a very brief description of the possibilities of Omola is presented here. The interested reader will have to consult e.g. [3], [4] or [11], where the former two contains very detailed descriptions of Omola and OmSim and the latter contains a description of how to model power systems using Omola. A simple beginner’s tutorial to Omola and OmSim is presented in [2].

**Using Omola as a modeling tool**

Omola provides the concept of object-oriented modeling, and each type of model is described as different classes, where the actual models are instantiations of the class descriptions. The concept of inheritance allows for a natural way of building a class library with re-usage of model code and good interpretation of the model structure. Modeling in Omola is basically done in two ways. First, one can simply use a common editor and do the model coding directly in the Omola language. More important is, however, the connection between Omola and the simulation tool OmSim which provides a nice graphical user interface. When classes of models and the model structure has been provided using basic Omola coding, it is easy to create a complex model using the mouse-driven graphical tools of OmSim. Interconnections between different models are easily possible to introduce using the graphical tools. OmSim also contains a good simulator which makes it possible to directly draw some conclusions of the constructed model using simulation.

The modeling environment OmSim and the modeling language Omola is often used for examination of a dynamic system with algebraic constraints. It is possible to detect such things as nominal stability during simulations with different parameter values in the model. Robust stability is, however, not possible to analyze directly in the OmSim environment. This is the main motivation behind the work described in this paper.
Omola is a powerful modeling tool and if all the possibilities of the described model matrices are used, many robustness problems can be analyzed using Omola, even though they are complex.

**The matrix generation of OmSim**

The OmSim environment is capable of creating some Matlab m-files, corresponding to the analyzed model. The original model has to be linear which of course is a major drawback of the analysis. The Matlab files provides a set of model matrices corresponding to the ones described in (5) of the following section. These files contains the explicit names of the parameters which were defined in Omola. This provides us with possibilities of addressing each and every parameter and thus we are able to draw conclusions of their appearance in the different matrices. The explicit parameter names and their values in the Omola model are of course provided in another Matlab m-file. Note that without this symbolic representation of the parameters in the model matrices we would not have been able to do the transformation of the following section.

4. **Transformation to an uncertainty feedback form**

The aim of this section is to present a general way of transforming a model given as a set of differential-algebraic equations with parametric uncertainties in the different model matrices to a form with extracted uncertainties ready for robustness analysis with the \( \mu \)-toolbox.

**Transformation without inputs and outputs**

Physical models are often stated in terms of algebraic and differential equations. If we exclude inputs and outputs, they can be written, after linearization, on the form

\[
\begin{align*}
E \dot{x} &= Ax + Fz \\
Gz &= Hx
\end{align*}
\]  

(2)

Here \( x \) can be viewed as a state variable. Provided that \( E \) and \( G \) are invertible, the \( z \)-variable can be eliminated and the equations can be rewritten on state space form as

\[
\dot{x} = E^{-1}(A + FG^{-1}H)x
\]

However, it is often advisable to avoid the matrix inverses and keep the equation on the original form. One reason is that the flexibility of the format (2) often allows the coefficients to be expressed linearly in terms of physical parameters. This is less common for the state space form.

With uncertain parameters appearing linearly in all the matrices \( A, E, F, G \) and \( H \), the equations can be rewritten as

\[
\begin{bmatrix}
\dot{x} \\
z
\end{bmatrix} = \begin{bmatrix}
R + \sum_{k=1}^{m} \delta_k R_k \\
S_k \delta_k T_k
\end{bmatrix} \begin{bmatrix}
\dot{x} \\
x
\end{bmatrix}
\]

(3)
where \( \delta_1, \ldots, \delta_m \in [-1, 1] \). The rank factorization \( R_k = S_k \delta_k T_k \) should be done with a minimal number \( r_k \) of columns in \( S_k \). Define

\[
\Delta = \text{diag}\{\delta_1, \ldots, \delta_m\} \quad S = [S_1, \ldots, S_m,]
\]

\[
T = \begin{bmatrix} T_1 \\ \vdots \\ T_m \end{bmatrix} \quad K(s) = \begin{bmatrix} I & 0 \\ s^{-1}I & 0 \\ 0 & I \end{bmatrix}
\]

Then the system equations can be rewritten as

\[
\begin{bmatrix} \dot{x} \\ z \\ v \end{bmatrix} = \begin{bmatrix} R & S \\ T & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ \Delta v \end{bmatrix} = \begin{bmatrix} R & S \\ T & 0 \end{bmatrix} \begin{bmatrix} K(s) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{x} \\ z \\ \Delta v \end{bmatrix}
\]

Closing the \( [\dot{x}, z] \)-loop gives the equation \( v = M(s)\Delta v \) where

\[
M(s) = TK(s)[I - RK(s)]^{-1}S
\]

Hence the stability condition in Theorem (1) generalizes as follows.

**Proposition 1**

Suppose that the system (3) is Hurwitz stable for \( \delta_1 = \ldots = \delta_m = 0 \). Then it is stable for all \( \delta_1, \ldots, \delta_m \in [-1, 1] \) if and only if

\[
\mu \Delta (TK(i\omega)[I - RK(i\omega)]^{-1}S) < 1 \quad \omega \in [0, \infty]
\]

where \( \Delta \) is described by

\[
\Delta = \text{diag}(\delta_1, \ldots, \delta_m)
\]

**Transformation with inputs and outputs**

Consider a differential-algebraic system with inputs \( u \) and outputs \( y \) as (5).

\[
E\dot{x} = Ax + Bu + Fz \\
Gz = Hx + Ju \\
y = Cx + Du + K\dot{x} + Lz
\]

Performance analysis can now be done similar to the approach in the previous section. First write the system on the form

\[
\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = R \begin{bmatrix} \dot{x} \\ x \\ z \\ u \end{bmatrix} + \sum_{k=1}^{m} S_k \delta_k T_k \begin{bmatrix} \dot{x} \\ x \\ z \\ u \end{bmatrix}
\]

6
where $\delta_1, \ldots, \delta_m \in [-1, 1]$. Define $S$ and $T$ as before to get

$$
\begin{bmatrix}
\dot{x} \\
z \\
y \\
v
\end{bmatrix} = \begin{bmatrix} R & S \\ T & 0 \end{bmatrix} \begin{bmatrix}
\dot{x} \\
z \\
y \\
v
\end{bmatrix}.
$$

By changing the block partitioning and by introducing $K(s)$ as we did before we get

$$
\begin{bmatrix}
\dot{x} \\
z \\
y \\
v
\end{bmatrix} = \begin{bmatrix} \hat{R} & \hat{S} \\ \hat{T} & \hat{U} \end{bmatrix} \begin{bmatrix} K(s) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix}
\dot{x} \\
z \\
y \\
v
\end{bmatrix}.
$$

Closing the $[\dot{z}]$-loop gives an equation of the form

$$
\begin{bmatrix}
y \\
v
\end{bmatrix} = M(i\omega) \begin{bmatrix} u \\ \Delta v \end{bmatrix}
$$

where

$$
M(i\omega) = \hat{T}K(s)[I - \hat{R}K(i\omega)]^{-1}\hat{S} + \hat{U}
$$

Hence, Theorem 2, the performance theorem will become as follows.

**Proposition 2**

The performance specification $\|y\|_2 \leq \|u\|_2$ holds for all $u$ if and only if

$$
\mu_{\Delta P}(\hat{T}K(i\omega)[I - \hat{R}K(i\omega)]^{-1}\hat{S} + \hat{U}) < 1, \quad \omega \in [0, \infty[,
$$

where

$$
\Delta P = \{\text{diag}\{\Delta F, \delta_1 I_{r_1}, \ldots, \delta_m I_{r_m}\} : \sigma(\Delta F) \leq 1, \delta_1, \ldots, \delta_m \in [-1, 1]\}
$$

**Model reduction**

When introducing uncertainties in a differential-algebraic system as (5) the complexity of the system quickly becomes large. The rank factorization described earlier may still leave the model large and time-consuming to analyze.

In many cases it is possible to reduce the complexity of the model and still maintain a model that contains all the dominant features of the original system. For example, it might be possible to remove one of the uncertainties of the system if it does not affect the behavior much.

When doing the rank factorization numerically we consider a singular value decomposition where the rank of the system is equal to the number of non-zero singular values. If the singular value is small the impact of considering this singular value as zero will also be small. This procedure will reduce the rank and thus the dimension of the resulting matrix $M$. 

7
Other more elaborate reduction algorithms exist, such as e.g. multidimensional balanced realization, but the drawback with this is the computationally intense algorithms associated with it, based on LMIs, which makes the reduction procedure slow if the model which are to be reduced is large. The strength with the reduction method presented here is that it is fast. The rank factorization must still be done, so we lose no time comparing to an ordinary transformation. The main weakness is that we do not have a simple error bound of the model reduction as we might have had using other methods.

5. Uncertainty modeling for the Nordic Power System

The motivation of introducing computer tools for transforming an arbitrary model constructed with a modeling tool is that we are then able to analyze the robustness aspects of very large and complex systems. Perhaps the largest and most complex system made by man is the power network. Today, with the networks of many countries connected into a single large system with thousands of generators and power lines connecting them, there are problems of a magnitude impossible to analyze just a few years ago. With the rapid development of computer power it is now possible to analyze larger systems than was possible earlier. The standard approach for analyzing power systems has been to simulate an advanced model for different sets of parameters, in some kind of simulation software.

This section deals with the concept of making a robustness analysis of a linear power system model, presented in [1]. The model describes the Nordic power network system greatly simplified, with 16 generators and connections between them. The model is large enough to put strains on the computer power and to illustrate the problems associated with analysis of a large problem, but still small enough to grasp.

The robustness analysis consists of verification of robust stability for a given range of variations in the load description and the line description. The worst case parameters have been extracted during the analysis and the performance for these parameters has been compared with the worst gain possible. We also examine a case where we have disconnected a power line and we also do a simple reduction of some of the uncertain parameters.

First, a short presentation of how a power network model works is presented. Then the model of the Nordic power system is presented, and last a robustness analysis of the model is presented. This analysis is made in order to illustrate the possibilities of the computer tools, and it will not cover all aspects of a full robustness analysis of this complex model.

Modeling power systems

A power system model relies on the theory of synchronous generators and electric components. This theory will be omitted from the discussion here and the basic differential and algebraic equations will be described but not commented in detail. The model description is based on [1], [11] and [12].

A power system model consists basically of four different components

- generators, i.e. power plants
- a power grid consisting of transmission lines between the generators
- consumption loads
- controllers, which stabilizes and improves the prestanda
The power system should be able to provide power to the consumers at all times. This power should not vary in voltage and frequency too much. If some kind of variation occurs the system should be well damped in order to attenuate disturbances.

**Generator equations**

The generator can be modeled by three first-order differential equations and a set of algebraic equations. No effort has been made to explain the logic behind the generator equations. The interested reader should pursue this issue in a basic book on synchronous generators or for example [12].

\[
\frac{2H}{\omega_0} \frac{d\omega}{dt} = -\frac{D}{\omega_0} \omega + P_m - P_e
\]

\[
T_{d0}' \frac{dE_q'}{dt} = U_f - E_q' - (X_d - X_d')I_d
\]

\[
\frac{d\delta}{dt} = \omega - \omega_r
\]

\[
P_e = E_q'I_q + (X_q - X_d')I_dI_q
\]

\[
V_d = X_qI_q
\]

\[
V_q = E_q' - X_d'I_d
\]

\[
V_{d,g} = \frac{V_n}{V_B} (V_d \cos \delta - V_q \sin \delta)
\]

\[
V_{q,g} = \frac{V_n}{V_B} (V_d \sin \delta + V_q \cos \delta)
\]

\[
I_{d,g} = \frac{I_n}{I_B} (I_d \cos \delta - I_q \sin \delta)
\]

\[
I_{q,g} = \frac{I_n}{I_B} (I_d \sin \delta + I_q \cos \delta)
\]

The different variables and parameters are:

- \(\omega\)  mechanical rotor speed
- \(\delta\)  rotor angle
- \(\omega_r\)  reference speed of system
- \(\omega_0\)  synchronous frequency, here 50 Hz
- \(H\)  inertia time constant
- \(D\)  damping coefficient
- \(P_m\)  mechanical power
- \(P_e\)  electrical power
- \(T_{d0}'\)  d-axis transient time constant
- \(E_q'\)  voltage behind transient reactance
- \(U_f\)  generator field voltage
- \(X_d, X_q\)  d-axis and q-axis synchronous reactance
- \(X_d'\)  d-axis transient reactance
- \(I_d, I_q\)  current, d - and q-axis respectively
- \(V_d, V_q\)  voltage, d - and q-axis respectively
- \(I_{d,g}, I_{q,g}\)  global current, d - and q-axis respectively
- \(V_{d,g}, V_{q,g}\)  global voltage, d - and q-axis respectively
- \(V_n, I_n, V_B, I_B\)  scaling constants
Note that the internal variables are expressed in a local scale, and that the last four equations determines the voltages and currents as seen by the rest of the model. This is to get a decent scaling of the variables and parameters of the generator.

The power grid and the loads
The power grid serves the purpose of describing how the different generators influence each other. Basically the power grid, or network, consists of power lines between the generators. These lines are most easily modeled as a resistance and an inductance in series. The network is then a set of impedances connecting the different generators and loads.

The loads are also modeled as impedances. However, as the line impedance typically should transmit all the power without losses, the reactance is larger than the resistance. In the power loads the resistances are much larger than the reactances. In the ideal case we do not want any reactance at all as this introduces phase shifting in the load voltages. In practice it is impossible to have loads without any reactances, but they have to be small to avoid unpleasant effects in the system. Normally such reactances are compensated by capacitors or by using field voltage control of the generators. Both the lines and the loads are modeled by the impedance

\[ Z_l = R_l + iX_l. \]

The difference between the lines and the loads are that loads are considered to be connected to ground, thus implying a zero voltage at one end of the impedance.

Controllers
A power net system consisting of only generators, lines and loads does not have to be stable at all. Some kind of controller is introduced in order to stabilize the system and improve the performance. Traditionally this controlling of the power net introduces control of each single generator. This control is performed via the field voltage \( U_f \) and/or the mechanical power \( P_m \).

In the paper we have considered a simple controller structure which controls the voltages in the model. This controller stabilizes the system and makes it better damped. Lots of research effort has been put on power system stabilizers (PSS) which is another controller structure which further dampens the oscillative poles. Robustness of such controllers have been discussed in e.g. [5]. We have not used a PSS on our system but only the more common voltage controller.

A linearized model
As mentioned before, the transformation to the uncertainty feedback structure requires a linear model. In order to produce the eleven model matrices given in (5) we need to linearize the model. Nonlinearities appears in the generator equations, (7), which must be linearized in order to make the transformation.

By introducing an operating point and linearizing around this point we can easily get a linear model which can be used for analysis. The vital model features such as stability can still be analyzed because if the analysis of the linearized model gives robust stability then the nonlinear original model must also provide robust stability. The reverse conclusion is not true however.

This completes the description of the model equations. No Omola code has been provided in the paper, but the interested reader is recommended to read [11] or [12] which give detailed descriptions of Omola code for power nets.
The Nordic Power System model

The Nordel Power Network as presented by [1] has been used to illustrate a typical power network. This model describes the Nordel power system which connects the power nets of the Nordic countries Sweden, Norway, Denmark and Finland. The model consists of 16 generators and a total of 20 different transmission lines which connects these 16 generators. Power loads are introduced at 16 different places in immediate connection to the generators. This model is a major simplification of the actual power system which consists of hundreds of different generators and perhaps thousands of different transmission lines and at least as many different consumption points. In Figure 5 we see the model.

As we have seen, each generator contains three states and each controller contains one state, which implies 64 states total. One state corresponding to the absolute angles of the system is not observable. We consider the mechanical powers $P_m$ as inputs and electrical effects $P_e$ as outputs. This will leave us with 63 states, 16 inputs and 16 outputs in the model we have chosen to analyze. The need for computer tools in order to do this analysis is obvious.

The concept of the paper has been to analyze the effects of uncertainties in different parameters which appears linearly in the eleven model equations (5). In the 16 generator Nordel model we have very many different parameters, more than 500 different ones, even though some of them are not uncertain at all, such as scaling parameters. Furthermore, some of the parameters may be uncertain but cannot be analyzed as they appear in a nonlinear way in the model equations. This is a drawback of the transformation procedure presented in this paper but the interesting parameters of the power system model appears linearly.

The most apparent uncertainties in a power system model are of course the
loads. The loads vary in time and are impossible to predict exactly. Another source of uncertainty is the power lines. These connections over long distances typically vary in both reactance and resistance due to varying external conditions.

We have, as mentioned previously, 20 different lines. This will introduce 40 new uncertain parameters, adding up to a total of 72 parameters. This number of uncertain parameters is highly unpractical to analyze with the computer power available and only some of the lines are considered uncertain in the following analysis.

**Analysis**

A robustness analysis of a problem as large as this, is very time-consuming and a complete analysis is omitted here. The objective with this section is instead to show how the robustness routines in the $\mu$-toolbox and the transformation described can be used to analyze a complex problem.

First we analyze the system to see if robust stability is provided by certain sets of parameters. Then we extract the worst-case parameters for the discussed parameter set, and examine how large gain the model will have for these parameters. After this we analyze the effect on the robustness issues if we disconnect one of the transmission lines. Last we see if we can reduce the model to a simpler model, and we use the $\mu$-value to roughly compare the approximated model with the original one.

**Robust stability**

The first thing we must analyze in the system is if the model of the system still is stable when we introduce uncertainties of varying sizes in the model.

Let us assume that the different parameters may deviate 7% from their nominal values. We assume uncertain parameters in all the loads and in 6 of the lines. This introduces 44 different parameters if we consider the real and imaginary part of the impedances separately. A robust stability analysis yields the results presented in Figure 4. Only the upper bound of the $\mu$-value is displayed, because of bad convergence of the lower bound.

![Figure 4](image-url)

**Figure 4** Robust stability using real uncertainties. 7% deviation.

We see that the system is robustly stable for a 7% deviation, but it cannot handle deviations that are very much larger.

The time for computing a robust stability check like this extends to about 30 minutes on a fast computer when about 80 frequency points are analyzed.

**The worst parameters**

We have now discovered that our power system was robustly stable when exposed to parameter perturbations of 7% of the nominal values of the parameters. We would
also like to have an estimate of the parameters which gives us the worst performance. As the $\mu$ of the robust stability test is close to 1, the stability boundary for the chosen parameter set, this hints that there might be some serious performance degradation for some parameters. We would like to know these parameters so we can try to avoid them.

From the $\mu$-algorithm in the $\mu$-toolbox it is possible to extract the parameter combination which gives the worst behavior. The worst case parameters for the 7% deviation example was extracted, and the gain, \textit{i.e.} the norm of the transfer function was calculated for the original system and the worst-case system and the result can be seen in Figure 5.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure5}
\caption{Input-output gain of 16 generator model. Nominal system gain - lower solid line. Worst case parameter gain - dashed line. Maximum gain possible - upper solid line.}
\end{figure}

As we suspected the gain becomes very much higher than for the original system in a frequency range around 2 rad/s. The upper bound of the gain which we got from the function call implies that the function was fairly successful at finding the worst parameter combination in the frequency range.

\textbf{Disconnection of a line}

Robustness analysis is quite good at handling uncertainties of different kinds but if an uncertainty is of the kind that a parameter value becomes infinitely large it is not so easy to analyze. In our power system model we have such a case when \textit{e.g.} a tree falls on a transmission line and the line gets cut off. Such an uncertainty must be analyzed by a change in the model.

Let us for example consider a case where we disconnect the line between generators G5 and G11 in the model shown in Figure 5. This is in the Oslo-Gothenburg area and the example is not unrealistic as power lines in this area have suffered damages and been disconnected in recent years with power oscillations as a result. We assume the same parameter uncertainties as we have done in the previous examples, \textit{i.e.} 7% deviation from the original parameter values. Consider the robust stability $\mu$-plot in Figure 6.

As we see the system does not fulfill the basic robust stability criterion $\mu < 1$ anymore. Clearly the possible incident of trees falling on the power lines must be avoided or somehow taken care of to guarantee good behavior at all times for the power system.

\textbf{Model reduction}

As we have seen, computation time is often very large when using the robustness routines of the $\mu$-toolbox on large models. Earlier in the paper we discussed the need
for some kind of model reduction. A reduction will allow faster computations and might allow us to use simpler models in our pursuit of system understanding.

This reduction method does not take longer time than an ordinary building of the model, which is about 10 minutes for a model of the size of the power system model analyzed here. A tolerance of 0.01, i.e. the singular values less than 0.01 were considered to be zero gives us a reduced model where 22 of the parameters has been removed from the model. These excluded parameters consists of all the reactances from the loads and all the resistances from the power lines. Consider Figure 7.

In this figure we see that the $\mu$-plots for the two models are almost identical even though 22 of the parameters have been excluded from the model corresponding to the dashed line. The interpretation of this is that the model behaves almost as a model where the uncertainties of the power lines lies only in the reactance and the uncertainty of the power loads lies only in the resistance.

The ability of making a reduction like this is valuable as we might reduce the computation time when we analyze larger models, where we might use uncertainties in only some of the parameters and still get a realistic model.

6. Conclusions

Further details on the material presented in this paper can be found in [9].
A general way of transforming differential-algebraic systems with parametric uncertainty into a form ready for robustness analysis with the $\mu$-toolbox. The model can be done in the modeling language Omola and an automatic way of doing a robustness analysis has been implemented.

The routines are able to perform model reduction according to a simple algorithm. The main advantage of the reduction method is the speed of the reduction algorithm.

A large power system model of the Nordic Power System has been considered in the paper. Uncertainties in the loads and the transmission lines have been assumed, and some aspects of robustness analysis have been examined. An estimate of the worst-case parameters has been extracted from the system using $\mu$-analysis. Furthermore the effect of a removal of the transmission line between Gothenburg and Oslo, due to an accident has shown us that robustness analysis might be important to do in order to identify critical and sensitive points in a large model.

7. References


