Fault Detection and Model Quality Estimation Using Mixed Integer Linear Programming

Soheil Salehpour
Fault Detection and Model Quality Estimation Using Mixed Integer Linear Programming

Soheil Salehpour

Dept. of Computer Science and Electrical Engineering
Luleå University of Technology
Luleå, Sweden

Supervisors:
Professor Thomas Gustafsson
Associate Professor Andreas Johansson
To Ulrika and my parents
Robustness is a necessary property of a control system in an industrial environment, due to changes of the process such as changes of material quality, aging of equipment, replacing of instrument, manual operation (e.g. a valve that is opened or closed) etc. The uncertainties associated with the nominal process model is a concern in most approaches to robust control. The question is how to achieve a tight bound or shape of the uncertainty by using a set of measurement data. This active research area is known as model quality estimation.

Change detection is a quite active field, both in research and applications. Faults occur in almost all systems, and change detection often has the aim to locate the fault occurrence in time and to raise an alarm. Examples of faults in an industry are leakage of a valve, clogging of a valve or faults in measurement instruments.

A time-varying linear system is a realistic description of many industrial processes, and nonlinear behavior can then also be accounted for. Then, we can consider a linear system with time-varying parameters as the model uncertainty, e.g. an affine input-output approximation. Many time-varying changes or faults of industrial processes can be described as abrupt changes in parameters. The approach is to model them as piecewise constant parameters. The parameters of the linear time-varying system are then approximated for two purposes: 1) As uncertainty bounds for use in robust control. 2) Fault detection and isolation.

We present a method based on the assumption of piecewise constant parameters which results in a sparse structure of their derivative. A MILP (Mixed Integer Linear Programming) algorithm to maximize the sparsity of a matrix is introduced in this thesis.

We use the method to estimate the time-varying parameters of a blender’s hinged-outflow valve. This process is included in the pelletization of Luossavaara-Kirunavaara AB (LKAB) where the quality of iron ore pellets depends on many factors. One important issue is the mixing of binding material and slurry. The level of the blender is controlled by regulating a hinged-outflow valve. Then, the modelling of the valve is important, and the essential idea is to find a method to use the process model and the available measured data to detect two detrimental conditions and warn the operators. These two conditions are: 1) The hinged valve is coated with slurry and therefore has to be cleaned to maintain its function. 2) Slurry is improperly distributed so that it does not cover the outflow valve, which then loses its authority over outflow. The valve behaviour is nonlinear and depends on the viscosity of the materials in the tank. Therefore, we use the method to estimate the time-varying parameters of the valve. Simulation with measurement data from the LKAB facility at Malmberget, Sweden, shows the viability of the algorithm.
Then, we apply the method to the change in the mean model and compare it with four other change detection algorithms. Two applications, fuel monitoring and airbag control are treated with good results.

In another example, we consider a time-varying time-delay first-order process model. The gain, time-constant and time-delay are considered as uncertainties in this example. An estimate of the perturbations is produced based on the MILP method. The Padé-approximation and orthogonal collocation method are used to approximate the delay.

An overhead crane is used as an illustrative example, where the length of the pendulum, friction coefficient and the proportionality factor converting the control signal into the speed of the suspension point are time-varying and then considered as uncertainties and we try to estimate the bounds of these perturbations.
# Contents

## Part I

1. **Introduction**  
   - 1.1 Fault Detection, Uncertainties and Robust Control  
   - 1.2 Some Applications in Mineral Processing  
   - 1.3 Contribution in this Thesis  
   - 1.4 Future Research  

## References

13

## Part II

### Paper A

1. Introduction  
2. The Biaffine Input/Output System  
3. Disturbance Bound  
4. The $l_\infty$-optimization Method  
5. Sparse Matrix  
6. Method Based on MILP and the Derivative of Uncertainty  
7. Application to a Pendulum Process  
8. Simulation Results  
9. Conclusions and Future Work  

### References

34

### Paper B

1. Introduction  
2. Preliminaries  
3. Approximation of a Time-delay  
4. Application to a First-order Process Model with Time-delay  
5. Simulation Results  
6. Conclusions and Future Work  

### References

53

### Paper C

1. Introduction  
2. Preliminaries  

vii
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Simulation Results</td>
<td>60</td>
</tr>
<tr>
<td>4 Conclusions and Future Work</td>
<td>62</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>69</td>
</tr>
<tr>
<td>PAPER D</td>
<td>71</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>73</td>
</tr>
<tr>
<td>2 Bentonite Blending Process</td>
<td>74</td>
</tr>
<tr>
<td>3 Process Model</td>
<td>75</td>
</tr>
<tr>
<td>4 A MILP Algorithm for Change Detection</td>
<td>77</td>
</tr>
<tr>
<td>5 Experiments with Measured Data</td>
<td>80</td>
</tr>
<tr>
<td>6 Conclusions and Future Work</td>
<td>81</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>81</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

Several people have contributed to this project, directly and indirectly. First of all I would like to thank my super advisors Dr. Andreas Johansson and Professor Thomas Gustafsson for constant support and frequently answering for my never ending questions, and their inspiration and knowledge. I am grateful for the assistance from my contact persons at LKAB, Anders Björkman and Kjell-Ove Mickelsson without whom I would not be able to understand the ore pellet process. I appreciate the help from Dr. Johan Löfberg with improving the MILP algorithm. Also, I thank my dear friend and colleague Mr. Miguel Castaño and his co-workers Jesús García González and Pablo Fernández de Dios for drawing the tank process using their visualization toolbox. I am also thankful Dr. Stefan Rönnbäck at Optimization AB company for his brilliant knowledge and advice about the blending process and its detrimental conditions.

I thank my parents and Ulrika for their never ending source of love.

Soheil Salehpour
Luleå April 2009
The licentiate thesis work has been performed at the Division of Systems and Interaction of Luleå University of Technology in Sweden, under the supervision of Professor Thomas Gustafsson and Dr. Andreas Johansson. The motivation for the research was to investigate the uncertainty models and change estimation, particularly for the bentonite blending process of the ore pellet plant of LKAB.

This research project is financed by the Hjalmar Lundbohm Research Center (HLRC) funded by LKAB.

This thesis includes an introduction, to provide the necessary background for the reader, and the four following papers:


Part I
A structure is an assembly that serves an engineering function and a smart structure is one that serves this function smartly i.e. by responding adaptively in a pre-designed useful and efficient manner to changing environmental conditions [1]. The quality of iron ore pellets in the pelletization of LKAB depends on many factors, and changing conditions such as raw material quality and customer’s demands. Moreover, the process consists of several complicated processes, e.g. drilling, blasting, crushing etc, and therefore the term ”smart pellet” has been coined to capture this complexity.

Since a smart structure consists of many structural and control components, the possibility of failure is increased due to the complexity and existence of multiple sensors and actuators. Therefore, applications of model-based fault detection techniques have been studied for such active control systems, where the failures are assumed to occur in the sensors and actuators. Fault accommodation helps mitigate the effect of faults or age-dependant component degradation, and allows for the design of control systems which incorporate low-cost components. Typical faults in a process industry are e.g. aging of a pump, fault in an actuator such as a control valve or an additive error in the tacho of a DC motor. Other examples are leaks in a tank or pipe. All such faults have the potential to result in increased consumption of resources, reduced quality and interrupted production.

A control system is robust if it is insensitive to differences between actual system and the model of the system which is used to design the controller. These differences are referred to as model and plant mismatch or simply model uncertainty, and finding a mathematical representation of the model perturbation is a challenge of many researchers [2]. The time-varying delay of a conveyor belt, varying behaviour of a valve depending on different positions of opening, aging of parts etc, can be considered as uncertainties in process industry.

In this introduction, the theory of fault detection, uncertainties and change estimation is briefly described. A state-space system with uncertain time-varying parameters is discussed. Then, an overview of the iron ore process and specifically the bentonite blending process is given. Moreover, the data assembling of in- and out flow and control
signal of the bentonite blending process are explained. The model of an overhead crane and a first-order process model with time varying-delay are described. The contributions of the thesis are discussed in a separate section. Finally, future research and references are presented.

1.1 Fault Detection, Uncertainties and Robust Control

Fault detection and fault diagnosis becomes important for many technical processes to improve safety, reliability and increase the efficiency of supervision. The hardware redundancy based fault diagnosis, is one of the traditional diagnosis technique. The core of this scheme consists of the reconstruction of the process components using identical hardware components. The main advantage of this is its high reliability and the direct fault detection, but the use of hardware results in high costs and thus the application of this scheme is restricted to a number of key components. Signal processing schemes are based on the assumption that certain process signals carry information about the fault and fault detection can be achieved by a suitable signal processing method. These schemes are mainly used for processes in steady state, and their efficiency for detection of faults in dynamic systems is limited. The plausibility test is another approach and is based on the check of some simple physical laws, to provide information about the fault. The plausibility test is limited for detecting faults in a complex process or for isolating faults [3].

Due to these limitations of the classical approaches, model-based methods of fault detection are being developed since the early 1970s, see e.g. [4], [5], [6], [7], [8], by using input/output signals and dynamic process models. The idea is to validate a nominal process model using measurement data from the process. This is often done using a residual signal, which is supposed to be nonzero in the case of fault.

In the first decade of the history of the model-based fault detection various methods were developed. The framework of the model-based technique had been established and a survey paper was published in *Automatica 1990* [9]. The methods are classified after how the residual is generated

- Observer-based methods [4], [10]: The classical state observer is used for generating a residual.
- Parity space methods [11]: A straightforward model-based method of fault detection is to take a fixed model and run it parallel to the process, and generate the derivatives of nominal and process outputs as residuals.
- Parameter identification [7]: Since in most practical cases the process parameters are not known or partially not known, the parameters are estimated using input/output data and the dynamic of the process. The residual is the difference between the estimated parameters and some nominal parameters.
Knowledge about the uncertainties associated with the nominal process model is required in most approaches to robust control, and estimating bounds for the uncertainties from measurement data is known as model quality estimation. In most existing approaches, the true system, see e.g. [12], [13], [14], [15], [16], is a linear time-invariant system (where uncertainty is considered in both $H_\infty$ and $L_1$). A linear time-varying system is a more realistic assumption on the true process, but then the quality estimation problem becomes vary demanding, even for special structures such as coprime factor uncertainty [17].

A state-space system with affine dependence on the uncertain time-varying parameters is assumed in [18] as process description. This structure is used frequently in robust control and estimation ([19] and [20]).

The model can be obtained from three major sources.

- Human knowledge: the knowledge of operators and engineers about the process and how it reacts to different inputs. This can be e.g. logical rules and even constraints on process performance.

- Measured data: Input/output data is applied to obtain an analytical model, e.g. ARMAX model.

- Physical laws: Physical principles of the process is also used to obtain an analytical model.

The estimation of uncertainties is required to increase the robustness of control systems, and improve the “smart structure”. These models usually consist of a process, actuators and sensors for the control and the supervision purpose, and may describe nominal as well as faulty system behaviour. The mathematical models for linear dynamic systems include input-output description and state space representation.

It should be mentioned that the model uncertainty is not the only concern when it comes to robustness. Other considerations include sensor and actuator failures, physical constraints, changes in control objective, the opening and closing of loops, variation in the raw material etc. Furthermore, the numerical design algorithms may not be robust, and the real control problem may not be properly described.

The nonlinear and time-varying behavior of industrial processes can be modelled as time-varying perturbations. We will consider the real process as a linear system with time-varying parameters to decrease the complexity and to make it possible to solve the model validation and model quality estimation as linear optimization problems. This kind of process description is well suited when the process physics is known to some extent so that the model contains a limited number of unknown parameters. These models can take many forms, ranging from the first-order system with time-varying delay used for Ziegler-Nichols tuning of a PID-controller to large-scale models of e.g. a blast furnace or a paper pulp digester [21] and [22].

A state-space system with affine dependence on the uncertain time-varying parameters is considered in papers A and B to simulate the model-based uncertainties in an
overhead crane process and a first-order process with time-varying delay. The model structure can be expressed by using the Kronecker product $\otimes$ as \[18\]
\[
\begin{align*}
    x(k+1) &= Ax(k) + Bu(k) + P(\pi(k) \otimes x(k)) + D\pi(k) + Q(\pi(k) \otimes u(k)) \\
    y(k) &= Cx(k) + \eta(k) \quad (1.1)
\end{align*}
\]
where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{n \times q}$, $P \in \mathbb{R}^{n \times nq}$, $Q \in \mathbb{R}^{n \times qp}$ are constant matrices. The vector $x(k) \in \mathbb{R}^n$ represents the state, $y(k) \in \mathbb{R}^l$ is the measured output and $u(k) \in \mathbb{R}^p$ is the input while $\eta(k) \in \mathbb{R}^l$ is some additive disturbance. The vector $\pi(k) \in \mathbb{R}^q$ represents the parameter uncertainties, i.e. the deviation from the nominal parameter values. Note that they are allowed to enter both linearly and bilinearly with the state $x$ as well as with the input $u$.

In papers C and D, a time-varying linear regression model is considered, which is suitable to formulate the model quality estimation as a linear optimization problem. A general regression model is
\[
y(t) = \phi(t)^T \theta(t) + e(t) \quad (1.2)
\]
where $\theta(t)$ is a piecewise constant vector between these time indices $k^n$, and $e(t)$ is some noise signal. For an ARX $(n_a, n_b, n_c)$ model [23]
\[
\begin{align*}
    \phi(t)^T &= [-y_{t-1}, ..., -y_{t-n_a}, u_{t-n_b}, ..., u_{t-n_b-n_c+1}] \\
    \theta(t)^T &= [a_1^n, a_2^n, ..., a_1^n, b_1^b, b_2^b, ..., b_1^n]
\end{align*}
\]
where $\theta$ is the time-varying parameters.

1.2 Some Applications in Mineral Processing

A brief description of the ore pellet and bentonite blending are given here.

1.2.1 Ore Pellet Process

Luossavaara-Kiirunavaara AB (LKAB) is the producer of iron ore pellets as high-quality raw material for use in steelmaking. The refinement from rock to pellets is called pelletization process and includes the following main steps. It starts with the drilling, blasting and initial crushing of the rock in the underground mines and releasing raw ore. The crude ore is transferred up to be further refined. Magnetic separators and crushers separate the iron ore from the tailings. Further refinement is achieved by grinding and cleaning of the iron ore. The resulting slurry is then mixed with a binding media (such as bentonite) and rolled into balls in drums. The green pellets are finally dried and sintered in a pelletizing machine (Fig. 1.1).

A narrow size distribution of the green pellets is considered an important quality measure of the product both for the pellet production process itself and the succeeding
1.2. SOME APPLICATIONS IN MINERAL PROCESSING

The crude ore is transferred
Drilling, Blasting and Crushing

Magnetic separators
and crusher

Tailing
Iron ore

Bentonite

Rock

Slurry

Balling

Blending Process

Grinding and cleaning

Magnetic, Separators and Crusher

Drying and Sintering

Pellet

Figure 1.1: The ore pellet process

reduction process in steelmaking [24]. Some properties of the incoming pellet feed, such as moisture content, fineness and wettability, varies the green pellet growth rate and consequently the size distribution. The size distribution can be controlled by adjusting the screen openings for the recycling load and for the onsize fraction in the balling process. It is also possible to influence the growth rate by varying the moisture content or the binder dosage. Increasing the binder dosage is known to decrease the green pellet growth rate, making the pellets smaller, while increasing the water content results in an opposite effect [25].

In the presence of the bentonite binder, green pellet wet strength is mainly determined by the viscous forces of the binder liquid and the time of mixing. Too short time gives a poor mixing between slurry and bentonite, while too long time produces seeds. Adequate control of the bentonite blending is thus essential, and the level of the tank must be kept constant to provide suitable and effective mixing of the materials. Moreover, overflow of the tank must be prevented. Therefore, the function of the hinged outflow-valve is investigated. We explore a method to use available measured data to warn for two detrimental conditions of the valve.

1.2.1.1 Bentonite Blending Process

The blending tank is showed in Fig. 1.2. The inflows to the tank (Fig. 1.3) are slurry, oversize-flow (the oversize return material of the balling process) and bentonite. Two
impellers agitates and distributes the material inside the tank both horizontally and vertically. Moreover, they scrape the wall and the bottom of the tank. A PI-controller keeps the level of the blender by steering the blender’s outflow-valve. The slurry- and oversize-flows are measured with scales on the incoming conveyor belts, and the sum of these flows are used as feedforward to compensate the change of mass-inflow to the tank (the inflow measurements are properly delayed to compensate for the transport time of the conveyors). The setpoint of the PI-controller is decided locally by operators. The level of the tank is measured with a load cell (Mass) and is sent as measured value to the controller, where the control signal steers the hinged valve opening. The outflow (Fig. 1.4) is not measured but can be estimated using the inflow, the level, control signal and the dynamics of the tank.

There are three modes for the control of the outflow-valve: control, manual and cleaning mode. In control mode, the PI-controller regulates the hinged valve and the level of the tank. In manual mode, the valve is steered locally by operator and in the cleaning mode, the hinged valve is closed and scraped with a blade from one of the impellers.

The cleaning mode is necessary because the hinged valve gets coated by clay slurry, which reduces the flow and necessitates the valve to be more open to pass a given outflow. We want to perform the cleaning mode only when necessary to prevent the unnecessary interruption of the operation. This is one of our challenges, to determine when the difference between the estimated and nominal parameter values depends on coating. Moreover, the tank is sometimes not controllable when the distribution of slurry is unfavourable, which often occurs when the level of the tank is lower than 30 percent, although the exact level depends on the viscosity of the slurry. The absence of slurry around the hinged valve in low levels cause a constant small outflow that does not depend on the openness of valve and the control signal. The detection of this state and alerting the operator to increase the level to cover the hinged valve, is another goal of this work.

1.2.1.2 Data Assembling of the Blender Process

The measured data of the blending process is collected by using data base ”loggbok” of LKAB, with the help of Anders Björkman. We have manually changed the hinged outflow-valve degree and assembled measurements of the inflows of the tank, the level of the tank, and valve opening degree.

During the work, the ore pellet process stopped several times. Investigation by operators (Fig. 1.6) showed that overheating of the hydraulic pump which steered the hinged outflow-valve caused the break of the process. We realized that the hinged valve had been fixed in a certain position during a long time when the oil circulation of the pump was malfunctioning. So, the fan cooler of the pump had not worked properly when we changed the degree of the hinged valve and the oil consequently circulated inside the pump, and caused the overheating. Anyway, we found a fault in the hydraulic pump and the broken part has hopefully been replaced with a new part. This can be an example of a fault that had been detected with model-based method.
1.2. SOME APPLICATIONS IN MINERAL PROCESSING

Figure 1.2: The bentonite blending tank

Figure 1.3: The slurry and oversize
Figure 1.4: The outflow of the tank

Figure 1.5: The bentonite blending process
1.2. Some Applications in Mineral Processing

1.2.2 A First-Order Process Model with Time-Delay

In process control, it is common to approximate the process with a first-order system with time-delay. This form is frequently used e.g. for the Ziegler-Nichol tuning of PID-controller. It is described as

\[ G(s) = \frac{K}{1 + Ts} e^{-\tau s} \]  

(1.3)

where \( K, T \) and \( \tau \) are respectively the gain, time-constant and time-delay of the process.

A time-varying state-space model is as

\[ \dot{y}_c(t) = -\frac{1}{T(t)} y_c(t) + \frac{K(t)}{T(t)} u(t - \tau(t)) \]

where \( y_c \) is measured output. The state-space form of \( u(t - \tau(t)) \) is approximated as

\[
\begin{align*}
\dot{x}_1(t) &= \frac{A_0}{\tau(t)} x_1(t) + \frac{B_0}{\tau(t)} u(t) \\
u(t - \tau) &= C_0 x_1(t) + D_0 u(t)
\end{align*}
\]

where \( A_0, B_0, C_0 \) and \( D_0 \) represent an approximate realization of a 1 second time-delay.

1.2.3 Overhead Crane Process

An overhead crane process is marginally stable and therefore difficult for the MILP algorithm to manage. This is the reason why we used it to illustrate the method. The crane process with the position of the load \( x_p \) as measurement \( y_c \) is considered. The
process is modeled as a pendulum where the horizontal velocity of the suspension point is proportional to the control signal $u$. With $x_s$ as the position of the suspension point the linearized process is formulated as

$$
\begin{bmatrix}
\dot{x}_p \\
\ddot{x}_p \\
\dot{x}_s
\end{bmatrix}
=\begin{bmatrix}
0 & 1 & 0 \\
-g/l & -c/m & g/l \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_p \\
\dot{x}_p \\
x_s
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
b
\end{bmatrix} u
$$

where $g$ is the acceleration of gravity, $l$ is the length of the pendulum, $c$ is a friction coefficient, and $b$ is the proportionality factor converting the control signal into the speed of the suspension point. The friction coefficient is not exactly known and will vary with the speed of the load. The length of the pendulum $l$ will also vary and we assume that the proportionality factor $b$ is poorly known and possibly time-varying.

1.3 Contribution in this Thesis

Six papers are published at both international and Swedish conferences in automatic control.

1.3.1 Model quality estimation

We present two methods for estimating upper bounds for the uncertainties ($|\pi(k)| \leq \rho \in \mathbb{R}^q$). The first method minimizes the $l_\infty$-norm of the uncertainty and its derivative, and the second method is based on the MILP method.

An optimization approach is presented to estimate the time-varying parameter uncertainty in state-space systems. Particularly, the case of time-varying time-delay uncertainty is addressed. A first-order system with time-delay is used as illustrative example. Two parameterizations of the perturbations and two approximation methods (Padé- and OC-approximations) are considered. The promising results of the MILP method shows that exploiting the non-randomness of the perturbations is a viable approach to the estimating time-varying parameter uncertainty.


1.3.2 Change detection

We present a method for detecting changes and estimating time-varying parameters in
AR(X) models. The method is based on the assumption of piecewise constant parameters
resulting in a sparse structure of their derivative. To illustrate the algorithm and its
performance, we apply it to the change in the mean model and compare it with four
other change detection algorithms. Two applications, fuel monitoring and airbag control
are treated with good results. The AR(X) change model shows good performance of the
method in two illustrative examples.


1.3.3 Fault detection of the bentonite blending tank

We apply the model estimation method based on MILP to use the process model and
the available measured data to detect two detrimental conditions of the valve and warn
the operators. These two conditions are: 1) The hinged valve is coated with slurry and
therefore has to be cleaned to maintain its function. 2) Slurry is improperly distributed
so that it does not cover the outflow valve, which then loses its authority over outflow.
The valve behaviour is nonlinear and depends on the viscosity of the materials in the
tank. Therefore, we use a change detection method based on Mixed Integer Linear
Programming to estimate the time-varying parameters of the valve. Simulations with
measurement data from the LKAB facility at Malmberget, Sweden, shows the viability
of the algorithm.


1.4 Future Research

The Ore pellet process: Modelling, simulation and assembling of data of other parts of
this process, e.g. cone crusher, can be a future work. The measurement instruments
of the slurry sometimes lose accuracy and must be calibrated. To detect a sensor fault
using the measured data is one interesting challenge.
The bentonite blending process: Investigation of the effect of viscosity on parameters in this process, and testing the model with more measured data with different viscosity and conditions is a subject of future work. We may also use a more advanced valve model, but then we need more variation of the control signal.

Parameter estimation and change detection: Explore other properties of the noise, e.g., that the noise is expected to be random and uncorrelated with the signal, and a more systematic approach to find parameter values of the algorithm. The algorithm should also be investigated further for coloured noise and nonlinear systems.

The modelling of the bentonite blending process in the standard form (1.2.3) is another future topic of our research.
References


Part II
Two Algorithms for Model Quality Estimation in State-Space Systems with Time-Varying Parameter Uncertainty

Authors:
Soheil Salehpour and Andreas Johansson

Reformatted version of paper originally published in:
Two Algorithms for Model Quality Estimation in State-Space Systems with Time-Varying Parameter Uncertainty

Soheil Salehpour and Andreas Johansson

Abstract

We present two methods to estimate bounds of parameter uncertainty in state-space systems. In the first method, we minimize the $l_\infty$-norm of the perturbation and its derivative. In the second method, an estimate of the perturbation is produced based on a quantized approximation of the uncertainty and the sparse structure of its derivative. Less sensitivity to increased noise and changed model parameters is achieved by the second method. We use an overhead crane as an illustrative example.

Keywords-Model quality estimation, time-varying parameter uncertainty, MILP, sparsity, perturbation, uncertainty, optimization.

1 Introduction

The uncertainties associated with the nominal process model is a concern in most approaches to feedback control. The question is how to achieve a tight bound or shape of the uncertainty by using a set of measurement data. This active research area is known as model quality estimation. In the existing approaches to model quality estimation, the true system, see e.g. [1],[2],[3],[4] and [5], is a linear time-invariant system (where uncertainty is considered in both $H_\infty$ and $L_1$). A time varying linear system is a more realistic assumption, since nonlinear behavior can then also be accounted for. However, model quality estimation of time-varying perturbations appears to be difficult. Here, we consider a linear system with time-varying parameters as the model uncertainty. The prime drawback of assuming a parametric, time-varying uncertainty description is its possible shortcoming for describing unmodeled dynamics. However, if the process physics is reasonably well-known, then the unmodeled dynamics can be limited to high frequencies, which then can often be described by a parametric model.

We will assume a state-space system with affine dependence on the uncertain time-varying parameters. This structure is frequently used in robust control and estimation [9] and [10], and can be expressed by using the Kronecker product as [6]

\[
\begin{align*}
    x(k+1) &= Ax(k) + Bu(k) + P(\pi(k) \otimes x(k)) + \\
           &\quad Dr(k) + Q(\pi(k) \otimes u(k)) \\
    y(k) &= Cx(k) + \eta(k)
\end{align*}
\]  

(1)
where ⊗ denote the Kronecker product and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{p \times n}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{n \times q}$, $P \in \mathbb{R}^{n \times q}$, $Q \in \mathbb{R}^{n \times p}$ are constant matrices. The vector $x(k) \in \mathbb{R}^{n}$ represents the state, $y(k) \in \mathbb{R}^{l}$ is the measured output and $u(k) \in \mathbb{R}^{p}$ is the input while $\eta(k) \in \mathbb{R}^{l}$ is some additive disturbance. The vector $\pi(k) \in \mathbb{R}^{q}$ represents the parameter uncertainties, i.e. the deviation from the nominal parameter values. Note that they are allowed to enter both linearly and and bilinearly with the state $x$ as well as with the input $u$.

Here, we will present two methods for estimating upper bounds for the uncertainties ($|\pi(k)| \leq \rho \in \mathbb{R}^{q}$). The first method minimizes the $l_{\infty}$-norm of the uncertainty and its derivative, and the second method is based on optimization of the sparseness of the derivative of the quantized perturbation.

An approximation of (1) is given in Section 2. In Section 3, we define a bound for the disturbance. An $l_{\infty}$-optimization method is presented in Section 4. A MILP (Mixed Integer Linear Programming) algorithm to minimize the sparsity of a matrix is introduced in Section 5, and in Section 6, we use the MILP-optimization method to estimate the perturbation. An overhead crane process is represented in Section 7, and the methods are applied to this process in Section 8. Section 9 gives some concluding remarks and directions for future work.

2 The Biaffine Input/Output System

In this section, we give an approximation of (1) whose output is affine in the input $u$ and perturbation $\pi$. It is derived in [11] as a 1’st order Taylor approximation of $y$ with respect to $\pi$ and may be expressed as

$$
\begin{align*}
\zeta(k+1) &= A\zeta(k) + Bu(k) \\
\xi(k+1) &= A\xi(k) + P(\pi(k) \otimes \zeta(k)) + D\pi(k) + Q(\pi(k) \otimes u(k)) \\
\hat{y}(k) &= C(\zeta(k) + \xi(k))
\end{align*}
$$

with initial state $\zeta(0) = x(0)$ and $\xi(0) = 0$. By defining the disturbance $\nu = r + \eta$, where $r$ denotes the linearization error, the output of (1) may be expressed as

$$
y(k) = \hat{y}(k) + \nu(k)
$$
Furthermore, by collecting the signals into vectors as

$$\begin{align*}
\Pi &= \begin{bmatrix} \pi(0)^T & \pi(1)^T & \cdots & \pi(N-1)^T \end{bmatrix}^T \\
Y &= \begin{bmatrix} y(0)^T & y(1)^T & \cdots & y(N)^T \end{bmatrix}^T \\
V &= \begin{bmatrix} \nu(0)^T & \nu(1)^T & \cdots & \nu(N)^T \end{bmatrix}^T
\end{align*}$$

(2) may also be formulated [11] as

$$Y = \Upsilon + \Xi \Pi + V \quad (4)$$

where $\Xi = \Omega + \Psi + \Phi$, and $\Omega$, $\Phi$, $\Psi$ and $\Upsilon$ are defined in (3).

3 Disturbance Bound

To bound the disturbance $\nu(k)$ we will use the window norm, which for continuous time signals is defined as [13]

$$\|\nu\|_\omega = \sup_{t \geq 0} \int_0^t \omega(t-\tau) |\nu(\tau)| \, d\tau$$

where $\omega(t)$ is nonnegative and bounded by an exponentially decreasing function. In [11]
a discrete time window norm is defined as follows

**Definition 1:** A window sequence is a sequence \( \omega \in \mathbb{Z}^+ \to \mathbb{R}^+ \), which is not identically zero and satisfies \( \omega(k) \leq c e^{-ak} \) for all \( k \geq 0 \) and some positive \( c \) and \( a \).

Given a window sequence \( \omega(k) \), the window norm for a discrete time signal \( v(k) \) may then be defined as [11]

\[
\|v\|_\omega = \sup_{k \geq 0} \sum_{i=0}^k \omega(k-i) |v(i)|
\]  

(5)

It is remarked that (5) satisfies all properties of a norm, but the proof is omitted.

It is straightforward to see that the window norm is equal to the \( l_\infty \)-norm by choosing \( \omega \) as the unit pulse function. The drawback of \( l_\infty \)-norm is that it only considers the peak value of the signal without any averaging and may therefore be conservative. This problem is handled in the window norm by choosing a window function that averages over a suitable time interval, i.e. a pulse function with non-unit duration or a decaying exponential function. The window norm actually approaches the \( l_1 \)-norm by letting \( \omega \) approach a unit step function.

An assumption on the disturbance \( \nu = [\nu_1, ..., \nu_l] \) may now be expressed as

\[
\|\nu_j\|_\omega \leq \epsilon_j, j = 1, ..., l
\]  

(6)

We assume that each \( \omega(k) \) is monotone decreasing for \( k > 0 \). Then, as showed in [11] for scalar \( \nu \), the condition (6) can be expressed as

\[
W |V| \leq 1_{N+1} \otimes \epsilon
\]  

(7)

where

\[
W = \begin{bmatrix}
\omega(0) & 0 & \cdots & 0 \\
\omega(1) & \omega(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\omega(N) & \omega(N-1) & \cdots & \omega(0)
\end{bmatrix}
\]

\[
\omega = \text{diag}(\omega_1, \omega_2, \ldots, \omega_l) \text{ and } \epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_l]^T
\]

**4 The \( l_\infty \)-optimization Method**

We assume that the perturbation is small and slowly varying and that the disturbance \( \nu \) is also small. Here, we try to find the tightest bound for the perturbation. The size of the perturbation \( \pi \) is measured with \( \sup_k \|F \pi(k)\|_\infty \) where \( \|\cdot\|_\infty \) is the \( l_\infty \) vector norm and \( F \) is a diagonal weight matrix. The time-derivative of \( \pi \) is approximated by \((\Delta \pi)(k) = \pi(k) - \pi(k-1)\) and its size is measured with \( \sup_k \|G \Delta \pi(k)\|_\infty \), where \( G \) is a diagonal weight matrix.
Scaling of the vectorized perturbation \( \Pi \) and its derivative \( \Delta \Pi = [\Delta \pi(0)^T \Delta \pi(1)^T \cdots \Delta \pi(N-1)^T]^T \) is accomplished by

\[
(I \otimes F) \Pi = [(F \pi(0))^T (F \pi(1))^T \cdots (F \pi(N-1))^T]^T \\
(I \otimes G) \Delta \Pi = [(G \Delta \pi(0))^T (G \Delta \pi(1))^T \cdots (G \Delta \pi(N-1))^T]^T
\]

The \( l_{\infty} \)-optimization problem is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \| (I \otimes F) \Pi \|_{\infty} + \| (I \otimes G) \Delta \Pi \|_{\infty} + h \epsilon \\
\text{subject to:} & \quad W |Y_m - T - \Xi \Pi| \leq 1
\end{align*}
\]

where \( Y_m = [y_m(0)^T y_m(1)^T \cdots y_m(N)^T]^T \), and \( y_m(k) \) is the measured outputs of (1) for \( k = 0, \cdots, N \) and \( h \) is a weight for the disturbance.

5 Sparse Matrix

In numerical analysis, a sparse matrix is a matrix populated primarily with zeros. The concept of sparsity is useful in complex systems and many application areas such as network theory. Huge sparse matrices often appear in science or engineering when solving partial differential equations.

One common approach to seeking a sparse description is based on \( l_1 \)-norm regularization [7] which produces an approximation with a sparse structure.

In this article, we present an exact solution of sparsity by using MILP (Mixed Integer Linear Programming) to minimize the number of non-zero elements in a matrix or vector.

We consider the logical variable \( \delta_{ij} \in \{0,1\} \) and the matrix \( S = (s_{ij})_{i,j=1\ldots N} \) to be related as

\[
\begin{align*}
\delta_{ij} &= 1 \iff s_{ij} \neq 0 \\
\delta_{ij} &= 0 \iff s_{ij} = 0
\end{align*}
\]

and we aim to minimize \( \sum_{i,j=1}^N \delta_{ij} \) for \( i, j = 1, \ldots, N \). First, we assume \( \delta_{ij} = \delta^1_{ij} + \delta^2_{ij} \) and (9) is re-written as

\[
\begin{align*}
(\delta^1_{ij}, \delta^2_{ij}) &= (1,0) \iff s_{ij} > 0 \\
(\delta^1_{ij}, \delta^2_{ij}) &= (0,0) \iff s_{ij} = 0 \\
(\delta^1_{ij}, \delta^2_{ij}) &= (0,1) \iff s_{ij} < 0
\end{align*}
\]

We establish a more practical link between logical and real variables by

\[
\begin{align*}
m \delta^1_{ij} - M \delta^2_{ij} &\leq s_{ij} \leq M \delta^1_{ij} - m \delta^2_{ij} \\
\delta^1_{ij} + \delta^2_{ij} &\leq 1
\end{align*}
\]
where \( m = \min_{i,j}(|s_{ij}|) \) and \( M = \max_{i,j}(|s_{ij}|) \). Then, a MILP for maximizing sparsity of \( S \) is formulated as

\[
\text{minimize} \quad \sum_{i,j=1}^{N}(\delta_{ij}^1 + \delta_{ij}^2) \\
\text{subject to:} \quad (10)
\]

Here, we also present an approximation method which is more practical in a numerical sense. This method instead minimizes the number of elements of a matrix which are larger than a certain threshold. If we consider \( \mu \) as threshold, the logical variables are changed into

\[
\delta_{ij} = 1 \leftrightarrow |s_{ij}| > \mu \\
\delta_{ij} = 0 \leftrightarrow |s_{ij}| \leq \mu
\]

Then, the inequalities in (10) are substituted in (11) by

\[
2\mu\delta_{ij}^1 - M\delta_{ij}^2 - \mu \leq s_{ij} \leq M\delta_{ij}^1 - 2\mu\delta_{ij}^2 + \mu \\
\delta_{ij}^1 + \delta_{ij}^2 \leq 1
\]

where \( M = \max_{i,j}(|s_{ij}|) + \mu \).

6 Method Based on MILP and the Derivative of Uncertainty

Quantization as a means of approximating signals is widely used in digital control and other areas. The differentiation of a quantized signal produces a lot of zero samples and thus, it has a sparse structure.

Our basic assumption is that the perturbations are due to deterministic physical phenomena and thus have a non-random nature. In particular, we assume that the perturbations can be approximated by quantization. Then, the following sparsity property of its derivative is used for estimating the uncertainties and bounds for them. With the definitions

\[
\sigma_i = [\delta_{1i}, \delta_{2i}, \cdots, \delta_{qi}] \quad \text{for} \quad i = 0, \cdots, N-1, \\
\Sigma_N = [\sigma_0, \sigma_1, \cdots, \sigma_{N-1}]^T, \quad U = I_{Nq} \otimes [M, -2\mu], \\
L = I_{Nq} \otimes [2\mu, -M] \quad \text{and} \quad \Lambda = I_{Nq} \otimes [1, 1]
\]

the sparsity constraint (12) applied to the derivative \( \Delta \Pi \) may be expressed as

\[
L\Sigma_N - 1_{Nq}\mu \leq \Delta \Pi \leq U\Sigma_N + 1_{Nq}\mu \\
\Lambda\Sigma_N \leq 1_{Nq}
\]

(13)
We use the constraint (13), and MILP-optimization is formulated as
\[
\minimize \quad \sum_{N=1}^{N_q} + \| (I \otimes F) \Pi \|_\infty \\
\text{subject to:} \quad (13) \\
W |Y_m - Y - \Xi \Pi| \leq 1_{N+1} \otimes \epsilon
\]
where we use the $\epsilon$ which is computed in (8) to increase the accuracy in this algorithm. Alternatively, $\epsilon$ may be computed using the sample $l_1$-norm method in [12].

7 Application to a Pendulum Process

We will apply the two methods to an overhead crane process with the position of the load $x_p$ as measurement $y_c$. The process is modeled as a pendulum where the horizontal velocity of the suspension point is proportional to the control signal $u$. With $x_s$ as the position of the suspension point the linearized process is formulated as
\[
\begin{bmatrix}
\dot{x}_p \\
\dot{x}_p \\
\dot{x}_s
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
-g/l & -c/m & g/l \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_p \\
\dot{x}_p \\
x_s
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
b
\end{bmatrix} u
\]
\[
y_c = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix}
x_p \\
\dot{x}_p \\
x_s
\end{bmatrix}^T
\]
where $g$ is the acceleration of gravity, $l$ is the length of the pendulum, $c$ is a friction coefficient, and $b$ is the proportionality factor converting the control signal into the speed of the suspension point. The friction coefficient is not exactly known and will vary with the speed of the load. The length of the pendulum $l$ will also vary and we assume that the proportionality factor $b$ is poorly known and possibly time-varying. We introduce the notations $\lambda = g/l$ and $\kappa = c/m$ which are thus functions of time.

A simple Euler approximation gives the following discrete-time system
\[
x(k+1) = \begin{bmatrix}
1 & h & 0 \\
-h\lambda(k) & 1 - h\kappa(k) & h\lambda(k) \\
0 & 0 & 1
\end{bmatrix} x(k) +
\begin{bmatrix}
0 \\
0 \\
hb(k)
\end{bmatrix} u_c(k)
\]
\[
y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(k)
\]
where $u_c(k) = u(kh)$ and $y(k) = y_c(kh)$ while $\eta(k)$ is supposed to capture both the continuous-time noise term $\eta_c(kh)$ and the error introduced by the approximation.

The sampling interval in our simulation is $h = 0.05$ seconds. Nominal values of $\lambda, \kappa$ and $b$ and time-varying deviations from these are introduced as
\[
b(k) = b_0 + \pi_1(k), \quad \lambda(k) = \lambda_0 + \pi_2(k), \quad \kappa(k) = \kappa_0 + \pi_3(k)
\]
Thus, with the definitions

\[ A = \begin{bmatrix} 1 & h & 0 \\ -h\lambda_0 & 1 - h\kappa_0 & h\lambda_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

\[ P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

the process may be expressed as (1).

Since the process is oscillating, the disturbance \( \eta \) is dominated by the oscillation frequency. The window function \( \omega \) in the norm \( \| \cdot \|_\omega \) is thus chosen as a pulse function whose duration is one half oscillation period to provide averaging of the disturbance. This means

\[ \omega(k) = \begin{cases} 1/35 & 0 \leq k \leq 35 \\ 0 & \text{otherwise} \end{cases} \]

## 8 Simulation Results

To give some idea of the performance of the algorithms, we apply them to the pendulum process as follows. The output of (1) and (2) with an added white noise disturbance \( \eta \) and the parameter perturbation is compared in Fig. 1 (right). The perturbations are also depicted in Fig. 1 (left) where \( \lambda_0, \kappa_0 \) and \( b_0 \) are chosen as 3.3, 0.25 and 2, respectively. The \( l_\infty \)- and MILP-optimization methods are implemented in the YALMIP software [8] and applied to the pendulum process. The weighting factors are chosen as \( F = G = \text{diag}(1/b_0,1/\lambda_0,1/\kappa_0) \) and \( h = 1/\|y_m - C\zeta\|_\omega \) in (8) for the \( l_\infty \)-method, and \( F = \text{diag}(1/b_0,1/\lambda_0,1/\kappa_0) \) in (14) for the MILP-method.

The result of a simulation for the two methods is depicted in Fig. 2 (\( \lambda_0 = 3.3, b_0 = 2, \kappa_0 = 0.25 \)), which shows that although \( l_\infty \)-method estimates a tighter bound for the perturbation \( \pi_2 \), the MILP-method exhibits a better tracking of the true perturbation. To illustrate the sensitivity of the methods to the magnitude of the perturbation, four test cases have been considered. The shape of the parameter perturbations of the four cases is depicted in Fig. 1 (left), but the amplitude is given by Table 1. It is seen that the \( l_\infty \)-method in many cases estimates tighter bounds for the perturbation but on the other hand frequently under-estimates the bound.

The sensitivity of the algorithms to the changing of the parameters \( \lambda_0, b_0 \) and \( \kappa_0 \) are showed in Table 2. The sensitivity of the methods to different noise levels is illustrated in Table 3 for different \( \epsilon \) which is the window norm of the white noise. As noticed in Tables 2 and 3, the estimated bounds for the \( l_\infty \)-method are more sensitive to the changing
of $\lambda_0$, $\kappa_0$, $b_0$ and the noise level $\epsilon$. The sensitivity of the first algorithm to different parameters may be accounted to the fact that it does not consider the properties of the perturbations, e.g. the assumption that the derivative is sparse. Furthermore, the estimated perturbations are very often oscillating (Fig. 2 to Fig. 6), which is caused by the $l_\infty$-norm inherent property to limit the upper-bound.

In the experiments so far, the algorithms have estimated 3 parameters ($\lambda$, $\kappa$ and $b$) and in the case of the $l_\infty$-method also noise. To test how the algorithms behave in a simpler case, i.e. with fewer perturbations, we isolate one or more perturbations. First, we repeatedly omit one of the perturbations. The results of simulations for the isolation of $\pi_1$, $\pi_2$ and $\pi_3$ are respectively depicted in Fig. 3 to Fig. 5, which shows tighter bounds for both methods, and better tracking of uncertainties for the MILP-method. We finally isolate two perturbations and estimate the third. Fig. 6 shows tighter bounds for the uncertainties in both methods and the best tracking for the MILP-method.

In purpose to check how the methods behave for perturbations with non-spiky derivative, we consider sinusoid perturbations. The results for the MILP- and $l_\infty$-algorithms are depicted in Fig. 7, which shows a good tracking of sinusoid uncertainties and tighter upper bounds for the MILP-method.

Even though the $l_\infty$-optimization method sometimes produces better bounds for the perturbations than the MILP-optimization method (Table 1), the less sensitivity to the changing of parameters (compare the results in Table 2), smaller sensitivity to the increasing of noise (Table 3), better tracking and estimation of perturbations in MILP method, and more realistic parameter estimates compared to the rapid change of the $l_\infty$-optimization method, makes the MILP method a better candidate for finding acceptable uncertainty bounds. However, the disadvantage of the MILP-optimization method is the longer run time of it in comparison with the $l_\infty$-optimization method.
Figure 2: Estimates of perturbations $\pi_1$, $\pi_2$ and $\pi_3$ in (a), (b) and (c), respectively. True perturbations (solid line), $l_\infty$-method (...) and MILP-method (-.-).
Figure 3: Estimates of perturbations $\pi_2$ and $\pi_3$ in (a), (b), respectively. True perturbations (solid line), $l_\infty$-method (...) and MILP-method (-.-).

Table 1: Estimated upper-bounds of perturbations for $\lambda_0 = 3.3$, $\kappa_0 = 0.25$ and $b_0 = 2$

<table>
<thead>
<tr>
<th>True upper-bound for perturbation</th>
<th>Upper-estimated bound for $l_\infty$-method</th>
<th>Upper-estimated bound for MILP-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[0.2,1,0,0.5]$</td>
<td>$[0.60,1.4,0.27]$</td>
<td>$[0.48,1.8,0.5]$</td>
</tr>
<tr>
<td>$[0.4,1,0,0.5]$</td>
<td>$[0.90,1.75,0.25]$</td>
<td>$[0.95,2.0,0.5]$</td>
</tr>
<tr>
<td>$[0.2,2,0,0.5]$</td>
<td>$[0.70,2.2,0.30]$</td>
<td>$[0.70,2.4,0.5]$</td>
</tr>
<tr>
<td>$[0.2,1,0,1.0]$</td>
<td>$[0.60,1.25,0.70]$</td>
<td>$[0.80,1.8,1.0]$</td>
</tr>
</tbody>
</table>
Figure 4: Estimates of perturbations $\pi_1$ and $\pi_3$ in (a), (b), respectively. True perturbations (solid line), $l_\infty$-method (…) and MILP-method (-.-).

Table 2: The sensitivity of upper-estimated bounds to different values of $\lambda_0$, $\kappa_0$ and $b_0$. True upper-bound is $[0.2, 1.0, 0.5]$

<table>
<thead>
<tr>
<th>$[\lambda_0, \kappa_0, b_0]$</th>
<th>Upper-estimated bound for $l_\infty$-method</th>
<th>Upper-estimated bound for MILP-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3.0, 0.25, 2.0]</td>
<td>[0.70, 0.70, 0.26]</td>
<td>[0.32, 0.90, 0.47]</td>
</tr>
<tr>
<td>[3.7, 0.25, 2.0]</td>
<td>[0.48, 1.90, 0.35]</td>
<td>[1.60, 1.50, 0.50]</td>
</tr>
<tr>
<td>[3.3, 0.20, 2.0]</td>
<td>[0.85, 1.50, 0.23]</td>
<td>[0.60, 1.50, 0.50]</td>
</tr>
<tr>
<td>[3.3, 0.15, 2.0]</td>
<td>[1.70, 3.80, 0.00]</td>
<td>[0.50, 1.30, 0.43]</td>
</tr>
<tr>
<td>[3.3, 0.25, 1.5]</td>
<td>[0.40, 1.60, 0.30]</td>
<td>[0.37, 1.38, 0.47]</td>
</tr>
<tr>
<td>[3.3, 0.25, 2.5]</td>
<td>[0.90, 1.60, 0.20]</td>
<td>[0.63, 1.70, 0.50]</td>
</tr>
</tbody>
</table>
Figure 5: Estimates of perturbations $\pi_1$ and $\pi_2$ in (a), (b), respectively. True perturbations (solid line), $l_\infty$-method (...) and MILP-method (-.-).

Table 3: The sensitivity of upper-estimated bounds to the changing of $\epsilon$, true upper-bound is $[0.2,1.0,0.5]$.

<table>
<thead>
<tr>
<th>Window norm of white noise</th>
<th>Upper-estimated bound for $l_\infty$-method</th>
<th>Upper-estimated bound for MILP-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>$[0.4,1.5,0.30]$</td>
<td>$[0.40,1.1,0.50]$</td>
</tr>
<tr>
<td>0.10</td>
<td>$[0.6,1.4,0.27]$</td>
<td>$[0.48,1.8,0.50]$</td>
</tr>
<tr>
<td>0.20</td>
<td>$[0.5,1.0,0.30]$</td>
<td>$[0.40,1.6,0.45]$</td>
</tr>
<tr>
<td>0.30</td>
<td>$[0.5,1.2,0.30]$</td>
<td>$[0.40,1.6,0.40]$</td>
</tr>
</tbody>
</table>
Figure 6: Estimates of one perturbation, when the other two perturbations are isolated. \( \pi_1, \pi_2 \) and \( \pi_3 \) for the two methods in (a), (b) and (c), respectively. True perturbations (solid line), \( l_\infty \)-method (...) and MILP-method (-.-).
Figure 7: The estimation of sinusoid perturbations $\pi_1$, $\pi_2$ and $\pi_3$ in (a), (b) and (c), respectively. True perturbations (solid line), $l_{\infty}$-method (...) and MILP-method (-.-).
9 Conclusions and Future Work

Two optimization methods are presented to estimate bounds of time-varying parameter uncertainty in state-space system. An overhead crane was used as illustrative example.

The promising results of MILP-method shows that exploiting the non-randomness of the perturbations is a viable approach to the estimating time-varying parameter uncertainty.

For future research, other properties of noise and perturbations may also be exploited, e.g. that the noise is expected to be random and uncorrelated with the perturbations.

Acknowledgment

The authors wish to thank Hjalmar Lundbohm Research Center (HLRC) funded by LKAB for financing this research.

References


Model Quality Estimation in Systems with Time-Varying Time-delay Uncertainty

Authors:
Soheil Salehpour and Andreas Johansson

Reformatted version of paper originally published in:
Model Quality Estimation in Systems with Time-Varying Time-delay Uncertainty

Soheil Salehpour and Andreas Johansson

Abstract

The problem of estimating bounds for time-varying parameter perturbations using measurement data is addressed. In particular, time-varying time-delay is considered. An estimate of the perturbation is produced based on a quantized approximation of the uncertainty and the sparse structure of its derivative. The Padé-approximation and orthogonal collocation method are used to approximate the delay. A first-order system with time-delay is used as an illustrative example. The gain, time-constant and time-delay are considered as uncertainties here.

Keywords- Model quality estimation, time-varying parameter uncertainty, MILP, sparsity, perturbation, uncertainty, optimization, time-delay.

1 Introduction

The uncertainties associated with the nominal process model is a concern in most approaches to feedback control. The question is how to achieve a tight bound or shape of the uncertainty by using a set of measurement data. This active research area is known as model quality estimation. In the existing approaches to model quality estimation, the true system, see e.g. [1], [2], [3] and [4], is a linear time-invariant system (where uncertainty is considered in both $H_\infty$ and $L_1$). A time-varying linear system is a more realistic assumption, since nonlinear behavior can then also be accounted for. However, model quality estimation of time-varying perturbations appears to be difficult. Here, we consider a linear system with time-varying parameters as the model uncertainty. The prime drawback of assuming a parametric, time-varying uncertainty description is its possible shortcoming for describing unmodeled dynamics. However, if the physics of process is reasonably well-known, then the unmodeled dynamics can be limited to high frequencies, which then can often be described by a parametric model.

We will assume a state-space system with affine dependence on the uncertain time-varying parameters. This structure is frequently used in robust control and estimation [8] and [9], and can be expressed by using the Kronecker product as [5]

$$
x(k + 1) = Ax(k) + Bu(k) + P(\pi(k) \otimes x(k)) + 
\quad D\pi(k) + Q(\pi(k) \otimes u(k))
\quad y(k) = Cx(k) + \eta(k)
$$

(1)
where \( \otimes \) denote the Kronecker product and \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times p}, C \in \mathbb{R}^{l \times n}, D \in \mathbb{R}^{n \times q}, P \in \mathbb{R}^{n \times nq}, Q \in \mathbb{R}^{n \times qp} \) are constant matrices. The vector \( x(k) \in \mathbb{R}^n \) represents the state, \( y(k) \in \mathbb{R}^l \) is the measured output and \( u(k) \in \mathbb{R}^p \) is the input while \( \eta(k) \in \mathbb{R}^l \) is some additive disturbance. The vector \( \pi(k) \in \mathbb{R}^q \) represents the parameter uncertainties, i.e. the deviation from the nominal parameter values. Note that they are allowed to enter both linearly and bilinearly with the state \( x \) as well as with the input \( u \).

In [14], we present two methods to estimate bounds of parameter uncertainty in state-space systems. The methods are based on estimates of the perturbations from measurement data. Since the number of perturbations is usually larger than the number of measurement signals \( (q > l) \), this requires additional assumptions on the uncertainty. In [14], we firstly assume that the perturbation and its derivative is minimal. Then, we suppose the perturbation is minimal and its derivative is maximally sparse. This later approach is proved to be particularly fruitful.

In discrete-time systems, a time-delay is described using extra state variables [15], which raises difficulties when considering a time-varying time-delay. Instead, we will use a sampled continuous-time approximation. Then, we use the method based on optimization of the sparseness of the derivative of the quantized perturbation to estimate the variation of the time-delay.

In Section 2, an approximation of (1) is given and a bound for the disturbance is defined. A MILP (Mixed Integer Linear Programming) algorithm to maximize the sparsity of a matrix is described and a method based on this algorithm estimates the perturbations. In section 3, we investigate the use of Euler discretization of a realization of the delay based on Padé- and OC- approximations. A first-order process model with time-delay which is widely used in industry is considered as an example in Section 4 and simulation results are given in Section 5. Section 6 gives some concluding remarks and directions for future work.

2 Preliminaries

The vector of dimension \( n \) where each element is 1 is denoted as \( 1_n = [1, 1, ..., 1]^T \) and \( I_n \) is the identity matrix of dimension \( n \).

2.1 The Biaffine Input/Output System

In this section, we give an approximation of (1) whose output is affine in the input \( u \) and perturbation \( \pi \). It is derived in [10] as a 1’st order Taylor approximation of \( y \) with respect to \( \pi \) and may be expressed as

\[
\begin{align*}
\zeta(k+1) &= A\zeta(k) + Bu(k) \\
\xi(k+1) &= A\xi(k) + P(\pi(k) \otimes \zeta(k)) + D\pi(k) + Q(\pi(k) \otimes u(k)) \\
\hat{y}(k) &= C(\zeta(k) + \xi(k))
\end{align*}
\]
with initial state $\zeta(0) = x(0)$ and $\xi(0) = 0$. By defining the disturbance $\nu = r + \eta$, where $r$ denotes the linearization error, the output of (1) may be expressed as

$$y(k) = \hat{y}(k) + \nu(k)$$

Furthermore, by collecting the signals into vectors as

$$\Pi = [\pi(0)^T \pi(1)^T \cdots \pi(N-1)^T]^T$$
$$Y = [y(0)^T y(1)^T \cdots y(N)^T]^T$$
$$V = [\nu(0)^T \nu(1)^T \cdots \nu(N)^T]^T$$

(2) may also be formulated [10] as

$$Y = \Upsilon + \Xi \Pi + V$$  \hspace{1cm} (3)

where

$$\Xi = \begin{bmatrix}
0 & 0 & \cdots & 0 \\
C\phi(0) & 0 & \cdots & 0 \\
CA\phi(0) & C\phi(1) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
CA^{N-2}\phi(0) & CA^{N-3}\phi(1) & \cdots & 0 \\
CA^{N-1}\phi(0) & CA^{N-2}\phi(1) & \cdots & C\phi(N-1)
\end{bmatrix}$$

$$\Upsilon = \begin{bmatrix}
C\zeta(0) \\
C\zeta(1) \\
\vdots \\
C\zeta(N)
\end{bmatrix}$$  \hspace{1cm} (4)

and $\phi(i) = P(I_q \otimes \zeta(i)) + Q(I_q \otimes u(i)) + D$ for $i = 0, \ldots, N - 1$.

### 2.2 Disturbance Bound

To bound the disturbance $\nu(k)$ we will use the window norm, which for continuous time signals is defined as [11]

$$\|\nu\|_\omega = \sup_{t_\omega \geq 0} \int_0^{t_\omega} \omega(t - \tau) |\nu(\tau)| \, d\tau$$

where $\omega(t)$ is nonnegative and bounded by an exponentially decreasing function. In [10] a discrete time window norm is defined as follows

**Definition 1:** A window sequence is a sequence $\omega : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$, which is not identically zero and satisfies $\omega(t) \leq c e^{-at}$ for all $k \geq 0$ and some positive $c$ and $a$. 
Given a window sequence \( \omega(k) \), the window norm for a discrete time signal \( \nu(k) \) may then be defined as

\[
\|\nu\|_\omega = \sup_{k \geq 0} \sum_{i=0}^{k} \omega(k - i) |\nu(i)|
\]  

(5)

It is remarked that (5) satisfies all properties of a norm, but the proof is omitted. It is straightforward to see that the window norm is equal to the \( l_\infty \)-norm if one chooses \( \omega \) as the unit pulse function. The drawback of \( l_\infty \)-norm is that it only considers the peak value of the signal without any averaging and may therefore be conservative. This problem is handled in the window norm by choosing a window function that averages over a suitable time interval, i.e., a pulse function with non-unit duration or a decaying exponential function. The window norm actually approaches the \( l_1 \)-norm by letting \( \omega \) approach a unit step function. An assumption on the disturbance \( \nu = [\nu_1, \ldots, \nu_l] \) may now be expressed as

\[
\|\nu_j\|_\omega \leq \epsilon_j, j = 1, \ldots, l
\]  

(6)

We assume that each \( \omega(k) \) is monotone decreasing for \( k > 0 \). Then, as showed in [10] for scalar \( \nu \), the condition (6) can be expressed as

\[
W|V| \leq 1_{N+1} \otimes \epsilon
\]  

(7)

where

\[
W = \begin{bmatrix}
\omega(0) & 0 & \cdots & 0 \\
\omega(1) & \omega(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\omega(N) & \omega(N-1) & \cdots & \omega(0)
\end{bmatrix}
\]

and \( \epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_l]^T \).

### 2.3 Sparse Matrix

In numerical analysis, a sparse matrix is a matrix populated primarily with zeros. The concept of sparsity is useful in complex systems and many application areas such as network theory. Huge sparse matrices often appear in science or engineering when solving partial differential equations.

One common approach to seeking a sparse description is based on \( l_1 \)-norm regularization [6] which produces an approximation with a sparse structure.

In this article, we present an exact solution of maximization of sparsity by using MILP (Mixed Integer Linear Programming) to minimize the number of non-zero elements in a matrix or vector.
We consider the logical variable $\delta_{ij} \in \{0,1\}$ and the matrix $S = (s_{ij})_{i,j=1,...,N}$ to be related as

$$\delta_{ij} = 1 \leftrightarrow s_{ij} \neq 0$$
$$\delta_{ij} = 0 \leftrightarrow s_{ij} = 0 \quad (8)$$

and we aim to minimize $\sum_{i,j=1}^{N} \delta_{ij}$ for $i, j = 1, \ldots, N$. First, we assume $\delta_{ij} = \delta_{ij}^1 + \delta_{ij}^2$ and (8) is re-written as

$$\begin{align*}
(\delta_{ij}^1, \delta_{ij}^2) &= (1,0) \leftrightarrow s_{ij} > 0 \\
(\delta_{ij}^1, \delta_{ij}^2) &= (0,0) \leftrightarrow s_{ij} = 0 \\
(\delta_{ij}^1, \delta_{ij}^2) &= (0,1) \leftrightarrow s_{ij} < 0
\end{align*}$$

The above link between logical and real variables can also be expressed by using MILP inequalities

$$\begin{align*}
m \delta_{ij}^1 - M \delta_{ij}^2 &\leq s_{ij} \leq M \delta_{ij}^1 - m \delta_{ij}^2 \\
\delta_{ij}^1 + \delta_{ij}^2 &\leq 1 \quad (9)
\end{align*}$$

where $m = \min_{i,j}(|s_{ij}|)$ and $M = \max_{i,j}(|s_{ij}|)$. Then, a MILP for maximizing sparsity of $S$ is formulated as

$$\begin{align*}
\text{minimize} & \quad \sum_{i,j=1}^{N} (\delta_{ij}^1 + \delta_{ij}^2) \\
\text{subject to:} & \quad (9)
\end{align*}$$

Here, we also present an approximation method which is more practical in a numerical sense. This method instead minimizes the number of elements of a matrix which are larger than a certain threshold. If we consider $\mu$ as threshold, the logical variables are changed into

$$\begin{align*}
\delta_{ij} &= 1 \leftrightarrow |s_{ij}| > \mu \\
\delta_{ij} &= 0 \leftrightarrow |s_{ij}| \leq \mu
\end{align*}$$

Then, the inequalities (9) are substituted in (10) by

$$\begin{align*}
2\mu \delta_{ij}^1 - M \delta_{ij}^2 - \mu &\leq s_{ij} \leq M \delta_{ij}^1 - 2\mu \delta_{ij}^2 + \mu \\
\delta_{ij}^1 + \delta_{ij}^2 &\leq 1 \quad (11)
\end{align*}$$

where $M = \max_{i,j}(|s_{ij}|) + \mu$ and $\mu$ is a small number.
2.4 Method Based on MILP and the Derivative of Uncertainty

Some assumptions on the perturbations are required in order to estimate them. We assume that the perturbations are small and that the disturbance \( \nu \) is also small. The size of the perturbation \( \pi \) is measured with \( \sup_k \| F\pi(k) \|_\infty \), where \( \| . \|_\infty \) is the \( l_\infty \) vector norm and \( F \) is a diagonal weight matrix. Scaling of the vectorized perturbation \( \Pi \) is accomplished by \( (I \otimes F)\Pi = [(F\pi(0))^T (F\pi(1))^T \cdots (F\pi(N-1))^T]^T \)

Our next assumption is that the perturbations are due to deterministic physical phenomena and thus have a non-random nature. In particular, we assume that the perturbation can be modeled as \( \pi = L\sigma \), where \( \sigma \) is a series of pulses, representing deterministic real-world events and \( L \) is some linear system, in our case simply an integrator. This means that the perturbations can be approximated by quantization. Quantization as a means of approximating signals is widely used in digital control and other areas. The differentiation of a quantized signal produces a lot of zero samples and thus, it has a sparse structure. The time-derivative of \( \pi \) is approximated by \( (\Delta \pi)(k) = \pi(k) - \pi(k-1) \). Now, with the definitions

\[
\sigma_i = [\delta^1_i, \delta^2_i, \cdots, \delta^\mu_i, \delta^q_i] \text{ for } i = 0, \cdots, N - 1,
\Sigma_N = [\sigma_0, \sigma_1, \cdots, \sigma_{N-1}]^T, \quad U = I_{Nq} \otimes [M, -2\mu],
L = I_{Nq} \otimes [2\mu, -M] \text{ and } \Lambda = I_{Nq} \otimes [1, 1]
\]

the sparsity constraint (11) applied to the derivative \( \Delta \Pi \) may be expressed as

\[
L\Sigma_N - 1_{Nq}\mu \leq \Delta \Pi \leq U\Sigma_N + 1_{Nq}\mu
\]

\[
\Lambda\Sigma_N \leq 1_{Nq}
\]

where \( \Delta \Pi = [\Delta \pi(0)^T \Delta \pi(1)^T \cdots \Delta \pi(N-1)^T]^T \). Also, let \( Y_m = [y_m(0)^T y_m(1)^T \cdots y_m(N)^T]^T \), where \( y_m(k) \) are the measured outputs of (1). We now use the constraint (12), and MILP-optimization to formulate

\[
\begin{align*}
\text{minimize} & \quad 1_{2Nq}\Sigma_N + \| (I \otimes F)\Pi \|_\infty + H\epsilon \\
\text{subject to:} & \quad (12) \\
W |Y_m - T - \Xi\Pi| & \leq \epsilon
\end{align*}
\]

where \( H \) is a weight for the disturbance.

3 Approximation of a Time-delay

The Padé- and orthogonal collocation-(OC-) approximation are commonly used to approximate a time-delay ([12] and [13]). If an approximation of a time-delay in state-space form is given, then we can realize another time-delay by time-scaling of the step-response
using the following proposition.

**Proposition:** Consider a linear system on state-space form

\[
\dot{x}(t) = Ax(t) + Bu(t) \\
y(t) = Cx(t) + Du(t)
\]

with step-response \( G(t) \). Then a linear system with step-response \( G(t/\tau) \) is given by

\[
\dot{x}(t) = \frac{A}{\tau} x(t) + \frac{B}{\tau} u(t) \\
y_c(t) = Cx(t) + Du(t)
\]  

(15)

**Proof:** See appendix

This means e.g. that if (14) is an approximation of a 1 second time-delay. Then, an approximation of the arbitrary time-delay \( \tau \) may be realized as (15). An Euler approximation of (15) with sampling interval \( h \) is

\[
x(k + 1) = (I + \frac{A}{\tau} h)x(k) + \frac{B}{\tau} h u(k)
\]  

(16)

Now, we explore the effect of the Euler discretizing on the frequency response and stability of the Padé- and OC-approximations. The poles of the discrete-time model (16) are the eigenvalues of \((I + \frac{A}{A} h)\) which depend on the choice of \( h, \tau \) and the order of the Padé- and OC- methods. A nominal delay of the model is assumed to be given and we often choose a large order to get better approximation of the delay with respect to phase difference. So, we try to find a value for the sampling-time which guarantees stability of (16).

The Bode-diagram and pole-zero map of the Padé- and OC- approximations for different sampling intervals with order 7 and \( \tau = 10 \) are depicted in Fig. 1 and Fig. 2. Fig. 1 shows a good correspondence between the continuous models and the discretized Padé- and OC- approximations for frequencies under 0.2. As showed in Fig. 2, the OC- and Padé- methods are unstable for sampling intervals over 0.3 and 0.7, respectively. The sampling interval in our simulations are chosen to be \( h = 0.2 \) seconds.

4 Application to a First-order Process Model with Time-delay

In process control, it is common to approximate the process with a first-order system with time-delay as

\[
G(s) = \frac{K}{1 + Ts} e^{-\tau s}
\]

where \( K, T \) and \( \tau \) are respectively the gain, time-constant and time-delay of the process. We will apply the methods to such a process with measurement \( y_c \), i.e.

\[
\dot{y}_c = -\frac{1}{T} y_c + \frac{K}{T} u(t - \tau)
\]
and the state-space form of \( u(t - \tau) \) is approximated as

\[
\dot{x}_1(t) = \frac{A_0}{\tau} x_1(t) + \frac{B_0}{\tau} u(t)
\]

\[
u(t - \tau) = C_0 x_1(t) + D_0 u(t)
\]

where \( A_0, B_0, C_0 \) and \( D_0 \) represent an approximate realization of a 1 second time-delay. An Euler approximation gives the following discrete-time system

\[
x_1(k+1) = (I + \frac{A_0 h}{\tau}) x_1(k) + \frac{B_0 h}{\tau} u(k)
\]

\[
x_2(k+1) = (1 - \frac{h}{T_0}) x_2(t) + \frac{K h}{T} (C_0 x_1(k) + D_0 u(k))
\]

\[
y(k) = x_2(k)
\]

(17)

where \( u(k) = u_c(\cdot k \cdot h) \) and \( y(k) = y_c(\cdot k \cdot h) \).

We introduce two forms for parameterization of the uncertainties \( (T, K \) and \( \tau). The first form is

\[
T(k) = T_0 (1 + \pi_1(k)), \quad K(k) = K_0 (1 + \pi_2(k))
\]

\[
\tau(k) = \tau_0 (1 + \pi_3(k))
\]

(18)

A 1st order Taylor approximation of the right-hand side of (17) with respect to \( \pi_1, \pi_2 \) and \( \pi_3 \) is

\[
x_1(k+1) \approx (I + \frac{A_0 h}{\tau_0} - \frac{A_0 h}{\tau_0} \pi_3(k)) x_1(k) + \frac{B_0 h}{\tau_0} \pi_3(k) u(k)
\]

\[
x_2(k+1) \approx (1 - \frac{h}{T_0}) x_2(t) + \frac{K_0 h}{T_0} (1 - \pi_1 + \pi_2) (C_0 x_1(k) + D_0 u(k))
\]

Thus, with the definitions

\[
A = \begin{bmatrix} I + \frac{A_0 h}{\tau_0} & 0 \\ \frac{K_0 C_0 h}{T_0} & 1 - \frac{h}{T_0} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{B_0 h}{\tau_0} + \frac{B_0 h}{\tau_0} \pi_3(k) \end{bmatrix}
\]

\[
P = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{K_0 C_0 h}{T_0} & 0 & 0 \\ \frac{K_0 D_0 h}{T_0} & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ \frac{K_0 D_0 h}{T_0} & 0 & 0 \\ -\frac{B_0 h}{\tau_0} & 0 & 0 \end{bmatrix}
\]

\[
D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ \frac{K_0 D_0 h}{T_0} & 0 & 0 \\ -\frac{B_0 h}{\tau_0} & 0 & 0 \end{bmatrix}
\]
the process may be stated as (1).

To avoid the Taylor approximation, we also consider the following parameterization of the uncertainty

\[
\frac{1}{T(k)} = \frac{1}{T_0} + \pi_1(k), \quad \frac{K(k)}{T(k)} = \frac{K_0}{T_0} + \pi_2(k) \\
\frac{1}{\tau(k)} = \frac{1}{\tau_0} + \pi_3(k)
\] (19)

Then (17) is re-written as

\[
x_1(k + 1) = (I + \frac{A_0 h}{\tau_0} + A_0 h \pi_3(k)) x_1(k) \\
+ (\frac{B_0 h}{\tau_0} + B_0 h \pi_3(k)) u(k)
\]

\[
x_2(k + 1) = (1 - \frac{h}{T_0} - h \pi_1) x_2(t) \\
+ h (\frac{K_0}{T_0} + \pi_2)(C_0 x_1(k) + D_0 u(k))
\]

With the definitions

\[
A = \begin{bmatrix}
I + \frac{A_0 h}{\tau_0} & 0 \\
\frac{K_0 C_0 h}{T_0} & 1 - \frac{h}{\tau_0}
\end{bmatrix},
B = \begin{bmatrix}
\frac{B_0 h}{\tau_0} \\
\frac{K_0 D_0 h}{T_0}
\end{bmatrix}
\]

\[
P = \begin{bmatrix}
0 & 0 & 0 & 0 & A_0 h & 0 \\
0 & -h & C_0 h & 0 & 0 & 0
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
Q = \begin{bmatrix}
0 & 0 & B_0 h \\
0 & D_0 h & 0
\end{bmatrix}
\]

the process may be expressed as (1).

5 Simulation Results

The MILP-optimization algorithm (13) is applied to the process (17) with uncertainties parameterized as (18) and (19). Padé- and OC- approximations are utilized to approximate the delay, i.e. provide \(A_0\), \(B_0\), \(C_0\) and \(D_0\) in (17). The time-varying parameters are depicted in Fig. 4 as thick, solid lines and their nominal parameter values \(T_0\), \(K_0\) and \(\tau_0\) are chosen 10, 5 and 10, respectively. The MILP-optimization method is implemented in the YALMIP software [7] and applied to the process. Weighting factors are chosen as \( F = I_3 \) and \( F = \text{diag}(T_0, K_0, \tau_0) \) for parameterization (18) and (19), respectively. We take \( H = 1/ \| y_m - C \xi \|_\omega \) for both forms.
First, we test the process with the step input in Fig. 3(a). The order of the Padé- and OC- approximations are chosen to be 7. The results are depicted in Fig. 4, which shows good tracking of the gain ($K$) and time-delay ($\tau$), but the algorithms do not detect the variation of the time-constant ($T$).

In order to improve the result, we consider a combination of sinusoidal inputs with frequencies under 0.2 (Fig. 3(b)). The simulation results are showed in Fig. 5. All the methods except when we use (19) in combination with the OC-approximation, detect the change of $T$, and the best tracking is achieved by using (19) in combination with the Padé-approximation. Wherever we use (19), a variation of $T$ is obtained when $K$ changes. The reason is that $K/T$ is a parameter in these cases, i.e. the variation of $K$ is compensated by a change of $T$.

To improve the result further, we increase the order of the Padé- and OC- approximations. The results for order 12 are showed in Fig. 6, and it can be seen that all methods detect the change of $T$, though we have more oscillation in $T$, $K$ and $\tau$. For input signal frequencies over 0.2 or higher order than 12 of Padé- and OC- method we get oscillating behavior of the time-delay approximation, resulting in deteriorated results.

6 Conclusions and Future Work

An optimization approach is presented to estimate the time-varying parameter uncertainty in state-space system. Particularly, the case of time-varying time-delay uncertainty is addressed. A first-order system with time-delay is used as illustrative example. Two parameterizations of the perturbations and two approximation methods (Padé- and OC-approximations) are considered.

The promising results of the MILP method shows that exploiting the non-randomness of the perturbations is a viable approach to the estimating time-varying parameter uncertainty.
Figure 2: Pole-zero map of (a) Padé- and (b) OC- approximations with order 7, $\tau = 10$ and $h = 0.1, 0.3, 0.5, 0.7$.

Figure 3: (y) Output of (1) (solid line), (u) input (- - -), nominal output of (2) (---), output of (2) (...). (a) Step input (b) A combination of sinusoidal input.

For future research, a more systematic way choose appropriate sampling-time and order of Padé- and OC-approximations will be investigated. To improve the estimation, other properties of noise and perturbations may also be exploited, e.g. that the noise is expected to be random and uncorrelated with the perturbations.

Acknowledgment

The authors wish to thank the Hjalmar Lundbohm Research Center (HLRC) funded by LKAB for financing this research.
Figure 4: Step input. Estimates of perturbations $\pi_1$, $\pi_2$ and $\pi_3$ in (a), (b) and (c), respectively. True perturbations (thick solid line), (18) and OC-approximation (---), (19) and Padé-approximation (solid line), (19) and OC-approximation (---), (19) and Padé-approximation (---).
Figure 5: Sinusoidal inputs. Estimates of perturbations $\pi_1$, $\pi_2$ and $\pi_3$ in (a), (b) and (c), respectively. True perturbations (thick solid line), (18) and OC-approximation (...), (18) and Padé-approximation (solid line), (19) and OC- approximation (- - -), (19) and Padé-approximation (- - -).
Figure 6: Sinusoidal inputs, 12th-order time-delay approximation. Estimates of perturbations $\pi_1$, $\pi_2$ and $\pi_3$ in (a), (b) and (c), respectively. True perturbations (thick solid line), (18) and OC-approximation (---), (18) and Padé-approximation (solid line), (19) and OC-approximation (- - -), (19) and Padé-approximation (- - -).
The impulse-response of (14) is
\[ g_1(t) = Ce^{At}B + D\gamma(t) \]
with corresponding step response
\[ G_1(t) = \int_0^t g_1(p)dp = CA^{-1}e^{At}B + D\theta(t) \]
where \( \gamma \) and \( \theta \) represent the Dirac impulse and the Heaviside step, respectively. The time-scaled \( G_1(t) \) is
\[ G_{\tau} = G_1(\frac{t}{\tau}) = CA^{-1}e^{At/\tau}B + D\theta(t) \]
and the corresponding impulse response is given by the time derivative
\[ g_{\tau}(t) = Ce^{A\frac{t}{\tau}}B + D\gamma(t) \]
which has the realization (15).

References


Parameter Estimation and Change Detection in Linear Regression Models Using Mixed Integer Linear Programming

Authors:
Soheil Salehpour, Andreas Johansson, Thomas Gustafsson

Reformatted version of paper accepted for publication in:
Parameter Estimation and Change Detection in Linear Regression Models Using Mixed Integer Linear Programming

Soheil Salehpour and Andreas Johansson, Thomas Gustafsson

Abstract

We present a method for detecting changes and estimating parameters in AR(X) models. The method is based on the assumption of piecewise constant parameters resulting in a sparse structure of their derivative. To illustrate the algorithm and its performance, we apply it to the change in the mean model and compare it with four other change detection algorithms. Two applications, fuel monitoring and airbag control are treated with good results. The AR(X) change model shows good performance of the method in two illustrative examples.

Keyword-change detection, change in the mean model, AR model, ARX model, MILP, sparsity, optimization.

1 Introduction

The area of change detection is a quite active field, both in research and applications. Faults occur in almost all systems, and change detection often has the aim to locate the fault occurrence in time and to raise an alarm. Another application of parameter change detection is estimation of perturbations [8]. Parameter uncertainties in linear regression models implies uncertain coefficients of the polynomials of the transfer function, i.e. the Kharitonov polynomial approach to robust control [9].

In [1] and [2], surveys are given over on-line and off-line formulations of single and multiple change point estimation. In the on-line method, multiple filters are used in parallel, where each one is matched to a certain assumption on the abrupt changes. Two off-line strategies are also proposed, one is based on Markov Chain Monte Carlo techniques, and the other approach is based on a recursive local search scheme.

In change point estimation, the goal is to find a sequence \( k^n = [k_1, k_2, \cdots, k_n] \) of time indices, where both the number \( n \) and the locations \( k_i \) are unknown, such that the signal or the model of the signal can be described as piecewise constant. A general signal model is the linear regression

\[
y(t) = \phi(t)^T \theta(t) + \epsilon(t)
\]  

where \( \theta(t) \) is a piecewise constant vector between these time indices \( k^n \), and \( \epsilon(t) \) is some
noise signal. For an ARX \((n_a, n_b, n_c)\) model \([10]\)
\[
\phi(t)^T = [-y_{t-1}, ... , -y_{t-n_a}, u_{t-n_b}, ... , u_{t-n_b-n_c+1}]
\]
\[
\theta(t)^T = [a_1^t, a_2^t, ... , a_{n_a}^t, b_1^t, b_2^t, ... , b_{n_b}^t]
\]
The special case of a change in the mean model is achieved by \(\phi(t) = 1\) where \(\theta\) is the mean of the signal.

The assumption that \(\theta(t)\) is piecewise constant gives a sparse structure of its derivative. We will use the sparsity in order to estimate \(\theta(t)\), using an algorithm developed in \([8]\). In Section 2, a bound for the noise is defined. A MILP (Mixed Integer Linear Programming) algorithm to maximize the sparsity of a matrix is described and a method based on this algorithm estimates the parameters. Simulation results are given in Section 3. Section 4 gives some concluding remarks and directions for future work.

2 Preliminaries

The vector of dimension \(n\) where each element is 1 is denoted as \(1_n = [1, 1, ... , 1]^T\). Vectors and matrices are marked with bold-face font.

2.1 Noise Bound

To bound the noise \(e(k)\) we will use the window norm, which for continuous time signals is defined in \([7]\) and for discrete time signals in \([6]\) as
\[
||e||_\omega = \sup_{k \geq 0} \sum_{i=0}^k \omega(k-i) |e(i)|
\]
where \(0 \leq \omega(k) \leq c a^k\) for \(c > 0\) and \(0 < a < 1\).

It is straightforward to show, using the techniques in \([7]\) that \((2)\) satisfies all properties of a norm. Clearly, both the \(l_\infty\)-norm and \(l_1\)-norm are special case of the window norm on a finite interval. The \(l_\infty\)-norm is given by choosing \(\omega\) as the unit pulse function, and the \(l_1\)-norm is obtained by letting \(\omega(k) = 1\) on the interval. The drawback of the \(l_\infty\)-norm is that it only considers the peak value of the signal without any averaging and may therefore give conservative noise bounds for signals with large spikes. The \(l_1\)-norm on the other hand, may give a conservative bounds for small persistent signals. These problems are handled with the window norm by choosing a window function that averages over a suitable time interval, \textit{i.e.} a pulse function with non-unit duration or a decaying exponential function. An assumption on the noise \(e\) may now be expressed as
\[
||e||_\omega \leq \epsilon
\]
where \(\omega(k)\) is nonincreasing for \(k > 0\). Let \(N\) be the number of samples. Then, as showed in \([6]\) for scalar \(e\) on \(1 \cdots N\), the condition \((3)\) can be expressed as
\[
W |E| \leq 1_N \epsilon
\]
where
\[
W = \begin{bmatrix}
\omega(1) & 0 & \cdots & 0 \\
\omega(2) & \omega(1) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\omega(N) & \omega(N-1) & \cdots & \omega(1)
\end{bmatrix}
\]
and \( E = [e(1), \cdots, e(N)]^T \).

2.2 Sparse Matrix

A sparse matrix is defined as a matrix populated primarily with zeros. The concept of sparsity is useful in complex systems and many application areas such as network theory. Huge sparse matrices often appear in science or engineering when solving partial differential equations.

One common approach to seeking a sparse description is based on \( l_1 \)-norm regularization [3] which produces an approximation with a sparse structure.

In this article, we use an exact solution of maximization of sparsity by using MILP (Mixed Integer Linear Programming) to minimize the number of non-zero elements in a matrix or vector [8].

We consider the logical variable \( \delta_{ij} \in \{0, 1\} \) and the matrix \( S = (s_{ij}) \) for \( i = 1, \ldots, P \) and \( j = 1, \ldots, Q \) to be related
\[
\delta_{ij} = 1 \iff s_{ij} \neq 0 \\
\delta_{ij} = 0 \iff s_{ij} = 0 \quad (5)
\]
and we aim to minimize \( \sum_{i=1}^{P} \sum_{j=1}^{Q} \delta_{ij} \). The above link between logical and real variables can also be expressed by using MILP inequalities
\[
-M \delta_{ij} \leq s_{ij} \leq M \delta_{ij} \quad (6)
\]
where \( M = \max_{i,j}(|s_{ij}|) \). Then, a MILP for maximizing sparsity of \( S \) is formulated as
\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{P} \sum_{j=1}^{Q} \delta_{ij} \\
\text{subject to:} & \quad (6) \quad (7)
\end{align*}
\]
Here, we also use an approximation method which may be more practical in a numerical sense. This method instead minimizes the number of elements of a matrix which are larger than a certain threshold. If we consider \( \mu \) as threshold, the logical variables are changed into
\[
\delta_{ij} = 1 \iff |s_{ij}| > \mu \\
\delta_{ij} = 0 \iff |s_{ij}| \leq \mu
\]
Then, the inequalities (6) are substituted in (7) by
\[
-M \delta_{ij} - \mu \leq s_{ij} \leq M \delta_{ij} + \mu \quad (8)
\]
where \( \mu \) is a small number.
2.3 Method Based on MILP and the Derivative of Parameters

The differentiation of a piecewise constant parameter produces a lot of zero samples and thus, it has a sparse structure. The time-derivative of $\theta$ is approximated by $(\Delta \theta)(k) = \theta(k) - \theta(k-1)$.

Then, the following sparsity property of its derivative is used for estimating the parameters [8]. With the definitions

$$
\sigma_i = [\delta_{i1}, \delta_{2i}, \cdots, \delta_{qi}]^T \quad \text{for} \quad i = 1, \cdots, N,
$$

$$
\Sigma_N = [\sigma_0, \sigma_1, \cdots, \sigma_{N-1}]^T
$$

where $q = n_a + n_b$, the sparsity constraint (8) applied to the derivative $\Delta \Theta$ may be expressed as

$$
-M \Sigma_N - 1_{Nq} \mu \leq \Delta \Theta \leq M \Sigma_N + 1_{Nq} \mu
$$

(9)

where $\Delta \Theta = [\Delta \theta(1)^T, \Delta \theta(2)^T, \cdots, \Delta \theta(N)^T]^T$.

Also, let $Y_m = [y_m(1)^T, y_m(2)^T, \cdots, y_m(N)^T]^T$, where $y_m(k)$ are the measured outputs of (1).

$\Theta = [\theta(1)^T, \theta(2)^T, \cdots, \theta(N)^T]^T$ and

$\Xi = \text{diag}(\phi(1)^T, \phi(2)^T, \cdots, \phi(N)^T)$. We now use the constraint (9), and MILP-optimization to formulate

$$
\begin{align*}
\text{minimize} & \quad \| \Sigma_N \|_1 + H \epsilon \\
\text{subject to:} & \quad \text{(9)} \\
W |Y_m - \Xi \Theta| & \leq \epsilon
\end{align*}
$$

(10)

where $H$ is a weight to normalize the noise.

3 Simulation Results

To give some idea of the performance of the MILP-optimization method, we apply it to the change in the mean model and the AR(X) model (1).

The window function $\omega$ in the norm $\| . \|_\omega$ is chosen as a pulse function with a duration time $D$, i.e.

$$
\omega(k) = \begin{cases} 
1/D & 0 \leq k \leq D \\
0 & \text{otherwise}
\end{cases}
$$

The power spectral density (PSD) of $y_m$ is estimated and the frequency interval where the PSD is positive is used as pass-band in a low-pass filter $G$ to calculate $\hat{y} = Gy_m$. The filtered measurement $\hat{y}$ is then used in calculating an appropriate noise weight $H = 1/\| y_m - \hat{y} \|_\omega$. 
The numerical results were obtained using the CPLEX [5] optimization tool, available in Matlab via the YALMIP [4] toolbox. We use a PC with Intel (2)core processor, 2 GHz in our simulations.

**Example 1:** We consider one abrupt change of magnitude 4 in the mean signal (Fig. 1(a), where the noise is Gaussian with variance 1 and a duration time $D$ of 10 samples is used. Tests with different $\mu$ are performed, which shows the change is not detected for $\mu \geq 0.02$. This is due to the upper bound of the derivative ($|\Delta \Theta| \leq 0.021_n$) which gives an unacceptable approximation of $\Theta$. This test shows that changing $\mu$ is not a good way to improve this method. In the next step, we consider a coloured noise

$$e(t) = (q - 0.9)^{-1}e_r(t)$$  (11)

where $e_r$ is a white Gaussian noise with variance 1. Fig. 1(b) still shows a good approximation of the mean change model for this noise.

The result is compared with four different change point estimations in the statistical literature [1], the Bayesian approach, the maximum likelihood approach and two non-parametric approaches. The mean square parameter error (MSE) vs. signal to noise ratio is showed for different variances of noise in Fig. 2(a). The mean of the signal after change is varied from 5 to 0 and the scaled mean square parameter error (SMSE) vs. signal to noise ratio is depicted in Fig. 2(b), where the MSE and SMSE are defined as follows.

$$MSE = \frac{\sum_{t=1}^{N} \|\theta(t) - \hat{\theta}(t)\|^2}{N}$$

$$SMSE = \frac{\sum_{t=1}^{N} \|\theta(t) - \hat{\theta}(t)\|^2}{\sum_{t=1}^{N} \|\theta(t)\|^2}$$

Fig. 2 shows that the MILP-optimization method results in smaller MSE and SMSE for higher signal to noise ratio. The run time of the MILP-algorithm is 0.3 s in comparison with 0.001 s, 0.006 s and 0.03 s for the Bayesian, ML and non-parametric methods respectively.

In the next simulation, there are two abrupt changes of magnitudes 3 and 4 respectively. The noise is Gaussian with variance 1 and $D = 5$. The result is depicted in Fig. 1(c), and the algorithm readily finds the change of the mean value and estimates its magnitude.

**Example 2:** Two illustrative applications are given here, fuel monitoring and airbag control [1]. The result of the MILP-optimization method is applied to the fuel consumption filtering problem with $D = 20$ and $D = 5$, is showed in Fig. 3. The results for the airbag control problem with $D = 400$ and $D = 100$ are showed in Fig. 4. The algorithm delivers plausible detections of the change in the mean in both applications. It can be seen that more changes are detected when $D$ is decreased.

**Example 3:** In purpose to test the algorithm on a general AR model, we consider a process with $n_a = 2$ and AR parameters showed in Fig. 5(a). With a white Gaussian
noise of variance 0.1, the output depicted in Fig. 5(b) is obtained. The method gives a good estimate of the parameters and their changes (Fig. 5(a), $D = 5$). The run time of the MILP-algorithm is 44 s in this example.

**Example 4:** The algorithm investigates the data of a human EEG signal which has a change in AR parameters [1]. A second order AR model is used here which shows a location of the change point of 428 in Fig. 6(a). In [1], the change point of 428 is computed with an forth order AR model, which shows that our algorithm detects the change of EEG with a lower order AR model but at the cost of a significantly longer run time (920 seconds compared to 1 second for the algorithm from [1], due to the long data set (750 samples).

**Example 5:** The method is applied to an ARX change model with $n_a = 2$ and $n_b = 1$, where the parameters are showed in Fig. 7(a). The input and output are depicted in Fig. 7(b). The parameter estimates with $D = 5$ are showed in Fig. 7(a) and it is clear that the changes are tracked well. The coloured noise (11) is applied to the model, and the algorithm detects the parameter changes in this case as well (Fig. 8).

In order to test the algorithm in the case when the assumption of piecewise constant parameters is violated, we apply it to the case of exponential and ramp shape of the parameter $b_1$. This is illustrated in Fig. 9 and Fig. 10 which show good tracking of this parameter.

**Example 6:** The method is also applied to a nonlinear ARX change model as follows

$$y(t) = -a_1^{-1}y(t - 1) - a_2^{-1}y(t - 2) + b_1^0 u(t - 1)^2 + b_2^0 u(t - 1) u(t - 2) + e(t)$$

where

$$\phi(t)^T = [-y_{t-1}, -y_{t-2}, u_{t-1}, u_{t-2}]$$

$$\theta(t)^T = [a_1, a_2, b_1, b_2]$$

With $D = 5$ and a white Gaussian noise of variance 0.1, the parameters and the parameter estimates are depicted in Fig. 11(a), which shows that this method may be appropriate for this nonlinear ARX change model. The input and output are depicted in Fig. 11(b). In the next simulation, the abrupt change of magnitude for $a_2$ is 0.2 instead of 0.5. Fig. 12 shows that the algorithm cannot detect this smaller change due to the correlation between $u(t - 1)^2$ and $u(t - 1)u(t - 2)$.

4 Conclusions and Future Work

An optimization approach is presented to estimate abruptly changing parameters in linear regression models. In particular, the mean of a signal and the parameters of an AR(X) model are considered. Two signals are used as illustrative examples and the results are compared with four other different point change detection algorithms. Two applications are tested here, fuel monitoring and airbag control. The method is also
Figure 1: The change in the mean model. The mean of the signal (dash-dotted), the estimated mean model (solid) and the measurement signal (dotted) (a) An abrupt change of magnitude 4 and white Gaussian noise with $\sigma^2 = 1$, the estimated signal with $\mu = 0.02$ (dotted) (b) An abrupt change of magnitude 4 and coloured noise (c) Two abrupt change of magnitudes 3 and 4.
Figure 2: (+) MILP-optimization method, (o) Bayesian approach, (square) ML approach, (diamond) and (triangle) are two nonparametric approaches. (a) MSE vs. signal to noise ratio. (b) SMSE vs. signal to noise ratio.

Figure 3: The fuel consumption measurement data (dashed), the estimated mean signal (solid). (a) $D = 20$ (b) $D = 5$
Figure 4: The airbag control measurement data (dashed), the estimated mean signal (solid). (a) $D = 400$ (b) $D = 100$

Figure 5: The AR change model with a white Gaussian noise. (a) The estimated parameters (solid) and true parameters (dashed), $a_1^t$ and $a_2^t$ starting at 1.5 and 1, respectively (b) The output (solid)
Figure 6: (a) The estimated parameters of an AR(2) model, for the human EEG signal in (b).

Figure 7: The ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$). (a) The estimated parameters (dashed) and true parameters (solid), $a_1 t$, $a_2 t$ and $b_1 t$ starting at 1.5, 0.8 and 0.5, respectively (b) The output (solid) and input (dash-dotted).
Figure 8: The ARX change model with the coloured noise (11). (a) The estimated parameters (solid) and true parameters (dash-dotted), $a_1^t$, $a_2^t$ and $b_1^t$ starting at 1.5, 0.8 and 0.5, respectively. (b) The output (solid) and input (dash-dotted).

Figure 9: The ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$) and the exponential parameter change. (a) The estimated parameters (solid) and true parameters (dashed), $a_1^t$, $a_2^t$ and $b_1^t$ starting at 1.5, 0.8 and 0.5, respectively. (b) The output (solid) and input (dash-dotted).
Figure 10: The ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$) and ramp shaped parameter change. (a) The estimated parameters (dashed) and true parameters (solid), $a_1^t$, $a_2^t$ and $b_1^t$, starting at 1.5, 0.8 and 0.5, respectively (b) The output (solid) and input (dash-dotted)

Figure 11: The nonlinear ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$). (a) The estimated parameters (dashed) and true parameters (solid), $a_1^t$, $a_2^t$, $b_1^t$ and $b_2^t$ ending at 1.5, 0.6, 1 and 1.2, respectively (b) The output (solid) and input (dash-dotted)
applied to both linear and nonlinear AR(X) change models in some examples. Other types of parameter changes (ramps and exponential functions) are tested to show the efficiency of the algorithm for non-piecewise constant parameters. The results shows good performance of the method. However, the disadvantage of this method is the computational complexity of the MILP-optimization. The results thus rely on a good MILP solver such as CPLEX. This also makes the method less suitable for time-critical applications such as airbag control.

For future research, other properties of the noise may also be exploited, e.g. that the noise is expected to be random and uncorrelated with the signal. A more systematic approach to find parameters $D$ and $H$ is also desirable. The algorithm should be investigated further for coloured noise and nonlinear systems.

Acknowledgment

The authors wish to thank the Hjalmar Lundbohm Research Center (HLRC) funded by LKAB for financing this research. Also, we appreciate the help from Dr. Johan Löfberg with improving the MILP algorithm.
References


Fault Detection of a Blending Tank Process Using Mixed Integer Linear Programming

Authors:
Soheil Salehpour, Andreas Johansson, Thomas Gustafsson

Reformatted version of paper submitted to:
IFAC Workshop on Automation in Mining, Mineral and Metal Industry, 14-16 October 2009, Vina Del Mar, Chile.
Fault Detection of a Blending Tank Process Using Mixed Integer Linear Programming

Soheil Salehpour and Andreas Johansson, Thomas Gustafsson

Abstract

The quality of iron ore pellets depends on many factors. One important issue is the mixing of binding material and slurry. It is assumed that proper mixing is achieved by keeping the material in the blender tank at an appropriate level to achieve a suitable dwelling time. The level of the blender is controlled by regulating the blender’s hinged outflow-valve. Then, the modelling of the hinged valve is important, and the essential idea of this article is to find a method to use the process model and the available measured data to detect two detrimental conditions of the valve and warn the operators. These two conditions are: 1) The hinged valve is coated with slurry and therefore has to be cleaned to maintain its function. 2) Slurry is improperly distributed so that it does not cover the outflow valve, which then loses its authority over outflow. The valve behaviour is nonlinear and depends on the viscosity of the materials in the tank. Therefore, we use a change detection method based on Mixed Integer Linear Programming to estimate the time-varying parameters of the valve. Simulations with measurement data from the LKAB facility at Malmberget, Sweden, shows the viability of the algorithm.

Keyword- Blending tank, hinged valve, parameter estimation, change detection, ARX model, MILP, sparsity, optimization.

1 Introduction

Luossavaara-Kiirunavaara AB (LKAB) is the producer of iron ore pellets as high-quality raw material for use in steelmaking. The pelletization process includes the following main steps. It starts with the grinding and cleaning of the iron ore. The resulting slurry is then mixed with a binding media (such as bentonite) and rolled into balls in drums. The green pellets are finally dried and sintered in a pelletizing machine.

A narrow size distribution of the green pellets is considered an important quality measure of the product both for the pellet production process itself and the succeeding reduction process in steelmaking [2].

Some properties of the incoming pellet feed, such as moisture content, fineness and wettability, varies the green pellet growth rate and consequently the size distribution. The size distribution can be also controlled by adjusting the screen openings for the recycling load and for the on-size fraction in the balling process. It is also possible to influence the growth rate by varying the moisture content or the binder dosage. Increasing the binder
dosage is known to decrease the green pellet growth rate, making the pellets smaller, while increasing the water content results in an opposite effect. [1].

In the presence of the bentonite binder, green pellet wet strength is mainly determined by the viscous forces of the binder liquid and the time of mixing. Too short time gives a poor mixing between slurry and bentonite, while too long time produces seeds.

Adequate control of the bentonite blending is thus essential, and the level of the tank must be kept constant to provide suitable and effective mixing of the materials. Moreover, overflow of the tank must be prevented. Therefore, the function of the hinged outflow-valve is investigated. We explore a method to use available measured data to warn for two detrimental conditions of the valve.

The article is composed as follows. First a description of the bentonite blending process is presented in Section 2 which is followed by a model of the process and the blender’s hinged-outflow valve in Section 3. In Section 4, an identification method based on parameter estimation and change detection is described. Experimental results for a real blender process is presented in Section 5. Some conclusions are drawn and suggestions for future work are given in Section 6.

2 Bentonite Blending Process

The inflows to the tank are slurry, oversize-flow (the oversize return material of the balling process) and bentonite. Two impellers agitates and distributes the material inside the tank both horizontally and vertically. Moreover, they scrape the wall and the bottom of the tank (Fig. 1).

A PI-controller keeps the level of the blender by steering the blender’s outflow-valve. The slurry- and oversize-flows are measured with scales on the incoming conveyor belts, and the sum of these flows are used as feedforward to compensate the change of mass-inflow to the tank (the inflow measurements are properly delayed to compensate for the transport time of the conveyors). The setpoint of the PI-controller is decided locally by operators. The level of the tank is measured with a load cell (Mass) and is sent as measured value (Fig. 1) to the controller, where the control signal steers the hinged valve opening. The outflow is not measured but can be estimated using the inflow, the level, control signal and the dynamics of the tank.

There are three modes for the control of the outflow-valve: control, manual and cleaning mode. In control mode, the PI-controller regulates the hinged valve and the level of the tank. In manual mode, the valve is steered locally by operator and in the cleaning mode, the hinged valve is closed and scraped with a blade from one of the impellers.

The cleaning mode is necessary because the hinged valve gets coated by clay slurry (Fig. 2), which reduces the flow and necessitates the valve to be more open to pass a given outflow. We want to perform the cleaning mode only when necessary to prevent the unnecessary interruption of the operation. This is one of our challenges, to determine when the difference between the estimated and nominal parameter values depends on coating.
Moreover, the tank is sometimes not controllable when the distribution of slurry is unfavourable, which often occurs when the level of the tank is lower than 30 percent, although the exact level depends on the viscosity of the slurry. The absence of slurry around the hinged valve in low levels cause a constant small outflow that does not depend on the openness of valve and the control signal (Fig. 3). The detection of this state and alerting the operator to increase the level to cover the hinged valve, is another goal of this work.

3 Process Model

The dynamic of the mixing tank is

\[ \dot{m}(t) = q_{in}(t) - q_{out}(t) \]  

where \( m \) is the mass inside the tank, \( q_{out} \) is the outflow and \( q_{in} \) is the sum of two delayed inflows and bentonite inflow, i.e. \( q_{in}(t) = q_{slurry}(t - T_1) + q_{oversize}(t - T_2) + q_{bentonite}(t) \) where \( T_1 \) and \( T_2 \) are respectively the approximated delays of the conveyor belts of the slurry and return oversize.

The behaviour of the hinged valve is nonlinear, and a nominal curve of outflow vs. control signal of it is showed in Fig. 4(a). It can be seen, that the hinged valve has a dead-zone which is due to the fact that it does not respond before the level of the control signal reaches a certain level (12.5 percent). Besides, the slurry does not run out if the
Figure 2: Bentonite blending process whenever the hinged valve is coated.

Figure 3: Bentonite blending process in the absence of slurry around the hinged valve in low levels.
opening degree is too small due to the high viscosity. In the saturation zone, the opening angle is so large that changes of it does not further influence the opening area. Since we do not have enough measurement data of \( q_{\text{out}} \) and \( u \) (control signal) in the dead-zone and saturation conditions, a linearized model of the valve characteristic is considered a sufficient approximation (Fig. 4(a)).

\[
C(u) = \theta_0 + \theta_1 u \tag{2}
\]

The parameters \( \theta_0 \) and \( \theta_1 \) will be identified using the MILP method in the sequel. So, the flow through the hinged valve can be expressed as

\[
q_{\text{out}} = (\theta_0 + \theta_1 u(t)) \sqrt{2gh(t)} \tag{3}
\]

where \( h(t) \) is the level of the tank. From the process model (1) it follows that

\[
q_{\text{out}}(t) = q_{\text{in}}(t) - \dot{m}(t) \tag{4}
\]

which integrated yields,

\[
\int_{t_0}^{t_1} q_{\text{in}}(\tau) \, d\tau - m(t_1) + m(t_0) = \int_{t_0}^{t_1} q_{\text{out}}(\tau) \, d\tau
\]

\[
= \left[ \int_{t_0}^{t_1} \sqrt{2gh(\tau)} \, d\tau, \int_{t_0}^{t_1} u(\tau) \sqrt{2gh(\tau)} \, d\tau \right] [\theta_0, \theta_1]^T
\]

The assumption is that the values of \( \theta_0 \) and \( \theta_1 \) changes when the valve gets coated. As seen in Fig. 4(a), we suppose that the main effect is a decrease in \( \theta_0 \), i.e. larger control signal is needed to pass the same amount of outflow.

Whenever the slurry does not cover the hinged valve, the outflow is mostly constant and does not depend on the control signal, therefore we suppose that the result is that \( \theta_0 \) is positive and almost constant and that \( \theta_1 \) is approximately zero (Fig.4(b)).

4 A MILP Algorithm for Change Detection

4.1 Change point detection in ARX model

In change point estimation, the goal is to find a sequence \( k^n = (k_1, k_2, \ldots, k_n) \) of time indices, where both the number \( n \) and the locations \( k_i \) are unknown, such that the signal or the model of the signal can be described as piecewise constant. A general signal model is the linear regression

\[
y(t) = \phi(t)^T \theta(t) + e(t) \tag{5}
\]

where \( \theta(t) \) is a constant vector in each interval \( k_{i-1} < t \leq k_i \) (\( \theta(t) \) is piecewise constant), and \( e(t) \) is some noise signal. For an ARX model

\[
\phi(t)^T = [-y_{t-1}, \ldots, -y_{t-n_y}, u_{t-n_u}, \ldots, u_{t-n_u-n_b+1}]
\]

\[
\theta(t)^T = [a_1^1, a_2^1, \ldots, a_{n_y}^1, b_1^1, b_2^1, \ldots, b_{n_u}^1]
\]
The special case of a change in the mean model is achieved by $\phi(t) = 1$ where $\theta$ is the mean of the signal. The assumption that $\theta(t)$ is piecewise constant gives a sparse structure of its derivative. We will use the sparsity in order to estimate $\theta(t)$, using the algorithm developed in [6].

4.2 Noise Bound

To bound the noise $e(k)$ we will use the window norm, which for discrete time signals is defined in [4] as

$$||e||_\omega = \sup_{k \geq 0} \sum_{i=0}^{k} \omega(k-i) |e(i)|$$

where $0 \leq \omega(k) \leq c a^k$ for $c > 0$ and $0 < a < 1$.

It is straightforward to show, using the techniques in [5] that (6) satisfies all properties of a norm. Clearly, both the $l_\infty$-norm and $l_1$-norm are special cases of the window norm on a finite interval. The $l_\infty$-norm is given by choosing $\omega$ as the unit pulse function, and the $l_1$-norm is obtained by letting $\omega(k) = 1$ on the interval. The drawback of the $l_\infty$-norm is that it only considers the peak value of the signal without any averaging and may therefore give conservative noise bounds for signals with large spikes. The $l_1$-norm on the other hand, may give a conservative bounds for small persistent signals. These
problems are handled with the window norm by choosing a window function that averages over a suitable time interval, i.e., a pulse function with non-unit duration or a decaying exponential function. An assumption on the noise $e$ may now be expressed as

$$\|e\|_\omega \leq \epsilon$$  \hspace{1cm} (7)

We assume that $\omega(k)$ is nonincreasing for $k > 0$. Let $N$ be the number of samples. Then, as showed in [4] for scalar $e$, the condition (7) can be expressed as

$$W |E| \leq 1_N \epsilon$$ \hspace{1cm} (8)

where $1_N = [1, 1, \ldots, 1]^T$, and

$$W = \begin{bmatrix}
\omega(1) & 0 & \ldots & 0 \\
\omega(2) & \omega(1) & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\omega(N) & \omega(N-1) & \cdots & \omega(1)
\end{bmatrix}$$

and $E = [e(1), \cdots, e(N)]^T$.

### 4.3 Sparse Matrix

A sparse matrix is defined as a matrix populated primarily with zeros. The concept of sparsity is useful in complex systems and many application areas such as network theory. Huge sparse matrices often appear in science or engineering when solving partial differential equations.

One common approach to seeking a sparse description is based on $l_1$-norm regularization [3] which produces an approximation with a sparse structure.

In this article, we use an exact solution of maximization of sparsity by using MILP (Mixed Integer Linear Programming) to minimize the number of non-zero elements in a matrix or vector [6].

We consider the logical variable $\delta_{ij} \in \{0, 1\}$ and the matrix $S = (s_{ij})$ for $i = 1, \ldots, P$ and $j = 1, \ldots, Q$ to be related as

$$\delta_{ij} = 1 \leftrightarrow s_{ij} \neq 0$$
$$\delta_{ij} = 0 \leftrightarrow s_{ij} = 0$$ \hspace{1cm} (9)

and we aim to minimize $\sum_{i=1}^P \sum_{j=1}^Q \delta_{ij}$. The above link between logical and real variables can also be expressed by using MILP inequalities

$$-M\delta_{ij} \leq s_{ij} \leq M\delta_{ij}$$ \hspace{1cm} (10)

where $M = \max_{i,j}(|s_{ij}|)$. Then, a MILP for maximizing sparsity of $S$ is formulated as

minimize $\sum_{i=1}^P \sum_{j=1}^Q \delta_{ij}$
subject to: \hspace{1cm} (10) \hspace{1cm} (11)
4.4 Method Based on MILP and the Derivative of Parameters

The differentiation of a piecewise constant parameter produces a lot of zero samples and thus, it has a sparse structure. The time-derivative of $\theta$ is approximated by $(\Delta \theta)(k) = \theta(k) - \theta(k - 1)$.

Then, the following sparsity property of its derivative is used for estimating the parameters [6]. With the definitions

$$\sigma_i = [\delta_{1i}, \delta_{2i}, \cdots, \delta_{qi}]^T$$

for $i = 1, \cdots, N$,

$$\Sigma_N = [\sigma_0, \sigma_1, \cdots, \sigma_{N-1}]^T$$

where $q = n_a + n_b$, the sparsity constraint (10) applied to the derivative $\Delta \Theta$ may be expressed as

$$-M\Sigma_N \leq \Delta \Theta \leq M\Sigma_N$$  \hspace{1cm} (12)

where $\Delta \Theta = [\Delta \theta(1)^T, \Delta \theta(2)^T, \cdots, \Delta \theta(N)^T]^T$. Also, let $Y_m = [y_m(1)^T, y_m(2)^T, \cdots, y_m(N)^T]^T$, where $y_m(k)$ are the measured outputs of (5), $\Theta = [\theta(1)^T, \theta(2)^T, \cdots, \theta(N)^T]^T$ and $\Xi = \text{diag}(\phi(1)^T, \phi(2)^T, \cdots, \phi(N)^T)$. We now use the constraint (12), and MILP-optimization to formulate

$$\begin{align*}
\text{minimize} & \quad \|\Sigma_N\|_1 + H \epsilon \\
\text{subject to:} & \quad (12) \\
& \quad W |Y_m - \Xi \Theta| \leq \epsilon
\end{align*}$$  \hspace{1cm} (13)

where $H$ is a weight to normalize the noise.

5 Experiments with Measured Data

We have performed two tests and applied the MILP algorithm to the measurement data. The window function $\omega$ in the norm $\|\cdot\|_\omega$ is chosen as a pulse function with a duration time $D$, i.e.

$$\omega(k) = \begin{cases} 
1/D & 0 \leq k \leq D \\
0 & \text{otherwise}
\end{cases}$$

where $H = 1$ in both estimations.

In the first experiment, the sample-time is 20 seconds and the data are low-pass filtered before the simulation (Fig. 5(a)) and $D = 30$ samples. During the first 400 seconds, the opening grade of the hinged valve is changed from 80 to 22.6 percent, and the level of the tank is low (Fig. 5(a))(the level value of the tank is expressed in percent and scaled with the maximum level). As depicted in Fig. 5(b) and 5(c), the estimated $\theta_0$ is positive and $\theta_1$ is a small value, which indicates that the slurry is distributed such that the hinged valve is not covered. Then, the level of the tank is increased after 400 seconds (20 samples) and the valve opening degree is changed between 22 and 50 percent. Here, we get some nominal values to compare with the parameters in the case when we need
a cleaning mode. It is noticeable that the estimated $\theta_0$ decreases in this time-interval which agrees with a slowly increasing clay cover of the valve.

The effect of the low level of the tank is also investigated in the second test, where the level of the tank is low most of the time (Fig. 6(a)). The data are filtered to decrease the effect of noise and the sample-time is 100 seconds and $D = 90$ samples. The parameter $\theta_1$ is a low positive value after 70 sample where the level of tank is over 30 percent and otherwise almost zero or negative when the valve is not covered and is not steerable (Fig. 6(b) and 6(c)).

6 Conclusions and Future Work

The model of a blending tank process is investigated, and the parameters of the outflow-valve are estimated using a MILP algorithm. The purpose of this is to warn the operator whenever the hinged valve gets coated or, due to unfavorable distribution of the slurry, does not affect the outflow.

Investigation of the effect of viscosity on parameters is a subject of future work. Also, we will test the model with more measured data with different viscosity and conditions. We may also use a more advanced valve model, but then we need more variation of the control signal.

Modifying the algorithm to make it better suited for use in real time is another aspect.

Acknowledgment

The authors wish to thank the Hjalmar Lundbohm Research Center (HLRC) funded by LKAB for financing this research. Also, we appreciate the help from Mr. Miguel Castaño, Jesús García González and Pablo Fernández de Dios with drawing the tank process by using their visualization toolbox. We also thank Dr. Stefan Rönnbäck at Optimation AB company for his brilliant information and advice about the blending process and its detrimental conditions.

References

Figure 5: The hinged valve model change detection (a) The control signal (solid), the level of the tank (dash-dotted), the computed outflow using (4) (dotted), the estimated outflow (dashed), (b) Estimated $\theta_0$ (c) Estimated $\theta_1$. 
Figure 6: The hinged valve model change detection (a) The control signal (solid), the level of the tank (dash-dotted), the computed outflow using (4) (dotted), the estimated outflow (dashed), (b) Estimated $\theta_0$ (c) Estimated $\theta_1$. 
