NTNU

Norwegian University of
Science and Technology

Process Data Mining for Parameter Estimation

With the DYNIA Method

Arnt Ove Fordal

Master of Science in Engineering Cybernetics

Submission date: January 2010

Supervisor: Morten Hovd, ITK

Co-supervisor: Fredrik Dessen, Siemens AS

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

In existing oil and gas production systems one is often reluctant to introduce modifications to the control system. The reason is that this normally means that the production system has to be taken out of operational order to perform specifically designed experiments to identify the dynamics of the system.

However, a process can be identified from segments of process data that exploit the dynamics of the process so that obtaining a good model is achievable.

Dynamic Identifiability Analysis (DYNIA) is a method for assessment of and segment selections from process data. The DYNIA methodology has been studied in an earlier project. This master thesis will further investigate the method.

Tasks:

1) Account for earlier work done on the subject.
2) Review the DYNIA methodology in the context of data mining on process data.
3) Implement and test the DYNIA methodology. Document the implementation.
4) Investigate the use of the Differential Evolution Algorithm to enhance the functionality of DYNIA.
5) Perform a system identification case study with real process data where data mining by the DYNIA methodology is used.

Assignment given: 31. August 2009
Supervisor: Morten Hovd, ITK

Abstract

Updating the model parameters of the control system of an oil and gas production system for the reasons of cost-effectiveness and production optimization, requires a data set of input and output values for the system identification procedure.

A requirement for the system identification to provide a well performing model is for this data set to be informative.

Traditionally, the way of obtaining an informative data set has normally been to take the production system out of normal operational order, in the interest of performing experiments specificially designed to produce informative data. It is however desirable to use segments of process data from normal operation in the system identification procedure, as this eliminates the costs connected with a halt of operation.

The challenge is to identify segments of the process data that give an informative data set.

Dynamic Identifiability Analysis (DYNIA) is an approach to locating periods of high information content and parameter identifiability in a data set. An introduction to the concepts of data mining, system identification and parameter identifiability lay the foundation for an extensive review of the DYNIA method in this context.

An implementation of the DYNIA method is presented. Examples and a case study show promising results for the practical functionality of the method, but also raise awareness to elements that should be improved. A discussion on the industrial applicability of DYNIA is presented, as well as suggestions towards modifications that may improve the method.

Preface

This master's thesis is the result of the work and research done throughout the final term of the Master of Science degree in Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU).

The work counts as a continuation and an expansion of a specialization project performed during the fall of 2008. Both this thesis and the specialization project have been done in cooperation with Siemens AS. I am thankful for the opportunity to collaborate with this company.

I would like to thank my supervisors, professor Morten Hovd at NTNU and associate professor Fredrik Dessen at Siemens, for their help and guidance during the work. Dr. Tu Duc Nguyen at Siemens also deserves many thanks for the same reasons.

Stjørdal, January 20, 2009

Arnt Ove Fordal

Contents

Abstract i

Preface iii

Contents v

List of Figures vii

1 Introduction 1
- 1.1 Data Mining . . . 2
- 1.2 System Identification . . . 3
  - 1.2.1 The Data Set . . . 3
  - 1.2.2 Candidate Models . . . 4
  - 1.2.3 Assessment of the Candidate Models . . . 5
- 1.3 Identifiability and Uncertainty . . . 7
  - 1.3.1 Information Content . . . 8
  - 1.3.2 Persistence of Excitation . . . 9
  - 1.3.3 Uncertainty Analysis . . . 11
- 1.4 Outline of the Thesis . . . 14

2 Dynamic Identifiability Analysis - DYNIA 15
- 2.1 System Identification under Uncertainty . . . 16
- 2.2 The DYNIA Method . . . 18
  - 2.2.1 Monte Carlo Parameter Sampling . . . 22
  - 2.2.2 Constructing a Measure of Likelihood and Identifiability 24
  - 2.2.3 Moving Window Calculation of Likelihood and Identifiability Measures . . . 27

3 Implementing DYNIA 31
- 3.1 Examples . . . 34
  - 3.1.1 Example 1: First Order System . . . 34
  - 3.1.2 Example 2: Second Order System . . . 39
  - 3.1.3 Case study . . . 46

4 Industrial Applicability of DYNIA **53**
4.1 Window Size Selection 53
4.2 Computational Complexity 54
4.3 Practical Inclusion of DYNIA in a Logging System 55
4.4 Including DYNIA in the System Identification Procedure . . . 56

5 Modifications to DYNIA **57**
5.1 Parallel Computing 57
5.2 Improving the Parameter Sampling 58

6 Conclusion and Further Work **63**

References **65**

Appendix A: A Bayesian Approach to Identifiability **69**

Appendix B: Matlab Code **71**

List of Figures

1.1 The system identification procedure. Based on [18] 5
1.2 Sources of uncertainty. (Sayers et. al. 2002 [26]) 11
1.3 Scheme of a sampling-based sensitivity / uncertainty analysis. Courtesy of Andrea Saltelli 13

2.1 GLUE output uncertainty limits (from [33]) 20
2.2 The DYNIA procedure (Wagener et. al. 2003 [32]) 21
2.3 Example plot of SSE values corresponding to one specific parameter of Monte Carlo sampled parameter sets 22
2.4 SSE values of the behavioural set of Fig. 2.3 24
2.5 A parameter with a low uncertainty band and steep CLD, indicating high information content in the data set wrt. this parameter. The same parameter as in Fig. 2.3 26
2.6 A parameter with a wide uncertainty band and CLD with a gentle slope, indicating low information content in the data set wrt. this parameter 27
2.7 Plot of the CLD and gradients (eight ranges) for the same parameter as in Fig. 2.3 28
2.8 Plot of the information content of the same parameter as in Fig. 2.3 29
2.9 Parameter estimation with DYNIA for the same parameter as in Fig. 2.3. Darker color means larger gradient. The true parameter value is 1. 30

3.1 A scheme of the DYNIA implementation 32
3.2 Input/output data for this example. A first order system 34
3.3 Information content of the gain parameter 35
3.4 Parameter identifiability plot of the gain parameter 36
3.5 Information content of the time constant parameter 36
3.6 Parameter identifiability plot of the time constant parameter . . . 37
3.7 Information content of the time constant parameter after the second run 38
3.8 Parameter identifiability plot of the time constant parameter after the second run 38

3.9 Input/output data for this example. A second order system with time-delay 39
3.10 Information content with regards to the estimation of time delay . 40
3.11 Parameter identifiability plot of the time delay parameter 40
3.12 Information content of the first time constant parameter 41
3.13 Information content of the gain parameter 41
3.14 Information content of the second time constant parameter 42
3.15 Parameter identifiability plot of the second time constant 43
3.16 Information content of the first time constant on the second run of DYNIA 44
3.17 Parameter identifiability plot of the first time constant on the second run 44
3.18 Information content of the gain on the second run 45
3.19 Parameter identifiability plot of the gain on the second run 45
3.20 Input/output values of the data set used in the case study 47
3.21 Information content of the gain parameter 48
3.22 Parameter identifiability plot of the gain parameter 49
3.23 Information content of the time constant parameter 50
3.24 Information content of the time constant after the second run of DYNIA 50
3.25 Parameter identifiability plot of the time constant after the second run 51
3.26 Comparison between modelled and actual output. $T_1 = 2$, $k = 2$. 51

1 Introduction

System identification, in the wide sense, is the process of developing a mathematical representation (model) of the dynamic behaviour of the physical system at hand, based on observed data from the system. A model is required in modern monitoring and control to be able to control the system dynamics so that it acts accordingly to a desired behaviour.

Observed data is stored in a *data set*, connecting input and output values of the system. The data set can e.g. be obtained by performing experiments on the system, specificially designed to exploit the system dynamics such that the data set to a large extent describe the operational modes of the system.

In existing oil and gas production systems one is often reluctant to recalibrate the model parameters in order to introduce modifications to the control system, or to update the system model, for example to changes in reservoir and wells. The reason being that performing the specificially designed experiments mentioned above means taking the system out of operational order; a costly undertaking. The data set then has to be obtained from operational process data.

Process data from normal operation has many properties that can deteriorate the performance of the system identification procedure. Segments of missing data, extraordinarily disturbance in the measurements and long periods of stationary data that carry little information about the system dynamics are all properties that can render the data set useless for the solution of the system identification problem.

DYNamic Identifiability Analysis (DYNIA) has in an earlier project [10] been presented as a method that might have attributes that make it suitable to assess how the quality of a data set varies with time, thus potentially having the ability to identify segments where the data set show good properties with respect to the system identification procedure. In other words, DYNIA has been suggested as a method that can raise the threshold for when it is necessary to perform specifically designed experiments to obtain a data set rather than to collect data from normal operation.

In this master's thesis the DYNIA method will be further studied and reviewed, and its applicability towards the industry assessed. A Matlab implementation of the method will be presented, and relevant tests will be performed on this implementation. Possible modifications to the method will also

be discussed.

This chapter gives background information on data mining, system identification and identifiability, and concludes with an outline of the contents of the thesis.

1.1 Data Mining

Data mining is the extraction of implicit, previously unknown, and potentially useful information from data [34], something that has been done in one way or another for centuries. Examples of early formal methods of identifying patterns in data include Baye's theorem and regression analysis. The increasing power of computing technology has greatly increased data collection and storage. This has given need for a new generation of tools and techniques for automatic and intelligent analysis that find useful knowledge in the mountains of data that arise. These tools and techniques are the subject of the field of Knowledge Discovery in Databases (KDD) [9], a field that first and foremost encompass the process of finding interesting data. Generally, the process comprises three steps: raw data preprocessing, data mining, and interpretation of the results.

The preprocessing consists of assembling a target data set. The data mining will only uncover information and patterns that is present in the data set. Thus, the target data set must be selected so that the analyser can be certain that the wanted information actually is contained in the selected data. In effect, the target data set has to be large, but at the same time concise enough to reach an acceptable balance between computational demands and supplies.

With the KDD approach, the data mining will in the context of the problems considered in this thesis, consist of two classes of tasks; regression analysis and clustering. The reasoning for the regression analysis is for each time-series data point (i.e., input/output data at each time step), to find a model that reproduce the data, given the same input as the real system, with the least error. Following this, a clustering algorithm should, based on the results from the regression analysis, be able to classify the time-series data points according to how much information is uncovered at each data point (or more precisely, as shall be seen, the knowledgde uncovered at a number of data points before and after the data point being considered). In this way, the time-series data points will be classified by the amount of knowledge discovered at each time step. In effect, this yields the opportunity to identify segments of the data set that brings forth an unproportionate large amount of knowledge. This opportunity can intuitively be linked with the main problem considered in this thesis, to identify segments of the data set that show good properties with regard to the system identification procedure. Hence, the knowledge to be uncovered by the main data mining algorithm to be considered in this thesis, that is the beforementioned DYNIA, is precisely that; properties with regard

to the system identification procedure.

The final step of KDD is to verify the knowledge uncovered by the data mining algorithms. Since the result of the system identification procedure is a new or calibrated system model, this verification can be performed by testing the model performance, not only with the data set used by the data mining algorithm, but also with other data sets that the data mining algorithm was not trained on.

This introduction gives a general approach on how the fundamental theory of data mining and KDD can be used in the solution of problems like the ones herein to be considered. The next sections serve as introductions to important theoretical concepts that serve as a background to the specific data mining approach taken in this thesis.

1.2 System Identification

As was mentioned, system identification is the process of developing a model to describe the dynamics of the system at hand. More specificially, system identification deals with the selection of a model structure and a corresponding parameter set that give a high degree of similarity between the observed and simulated output data. This section serves as an introduction to the subject, and it is to a large extent based on the work performed by the author in [10]. [8] and [18] serve as excellent books on the subject.

According to [18], the system identification procedure consists of three basic entities:

1. A data set Z^N
2. A set of candidate models
3. A rule by which candidate models can be assessed using the data set

1.2.1 The Data Set

The data set Z^N consists of recorded input and output data. As shall be seen, the properties of the data set is detrimental to the quality of the overall solution of the system identification problem. The data set can either be obtained by performing an experiment on the physical system, or by collecting data from the normal operation of the system. In the former case, the engineer often has the opportunity to actuate the system with a predetermined input sequence designed to exploit a great part of the system dynamics, whereas in the latter case, collecting a prosperous data set Z^N is not necessarily a straightforward exercise. Collecting data by performing an experiment on the system is always preferable, but in running production systems one will often hesitate to implement modifications to the control system if it involves stopping the production to perform such experiments. Therefore, on many

occasions, it is preferable that one has the opportunity to rely on process data when performing the system identification procedure.

The DYNIA method has been presented as a method possibly giving the ability to obtain a prosperous data set from large amounts of process data, in the case where a big part of the process data exploit only small or inconclusible parts of the system dynamics. This is the background for the further review and analysis of DYNIA presented in this thesis.

1.2.2 Candidate Models

Specifying a set of candidate models is conceptually the most difficult part of the system identification procedure [6]. A model set is an unparametrized collection of models that share a certain property, e.g. a model set that contain all linear models. A model structure is the parametrization of a model, where each model in the set is indexed by the parameter vector Θ. The search for the 'best' model is performed on the model structure. [18]

A priori knowledge based on physical laws with regard to the system at hand, relationships to earlier experience of modeling of similar systems and litterature on the formal properties of models should all be taken into account when building the set of candidate models. Considerations has to be based on which kind of model is to be obtained, because mathematical models of petroleum production system can be divided into three groups [20]:

1. Theoretical models developed using chemical and physical principles
2. Empirical models obtained from mathematical analysis of process data
3. Hybrid models obtained as a combination of theoretical and empirical approach to model design

The reason for modelling a petroleum production system is usually that you want to estimate and/or control states that have a real physical meaning, such as pressures, flow rates, valve opening diameters etc. To achieve this, the most intuitive approach is to base the model sets on chemical and physical principles. However, empirical knowledge of the system is often used to reduce the complexity of the model, such that most practical models used in the petroleum production industry are hybrid, at least to some degree. Neural networks are often used as empirical model parts to describe complex parts and/or parts that do little to the overall dynamics of the system [14].

The problems to be addressed in this thesis are concerned with updates of the parameter set of a model after modifications has been made to the system itself, or to the control system. It shall on most occasions be assumed that the same model structure is to be used with the updated parameters, i.e. that the 'true' system representation is a part of the candidate set. When these assumptions are not made, model structure identification is used to evaluate

the underlying assumptions of the model structure, e.g. if the model structure is capable of reproducing different dominant modes of the dynamic behaviour with a single parameter set [32].

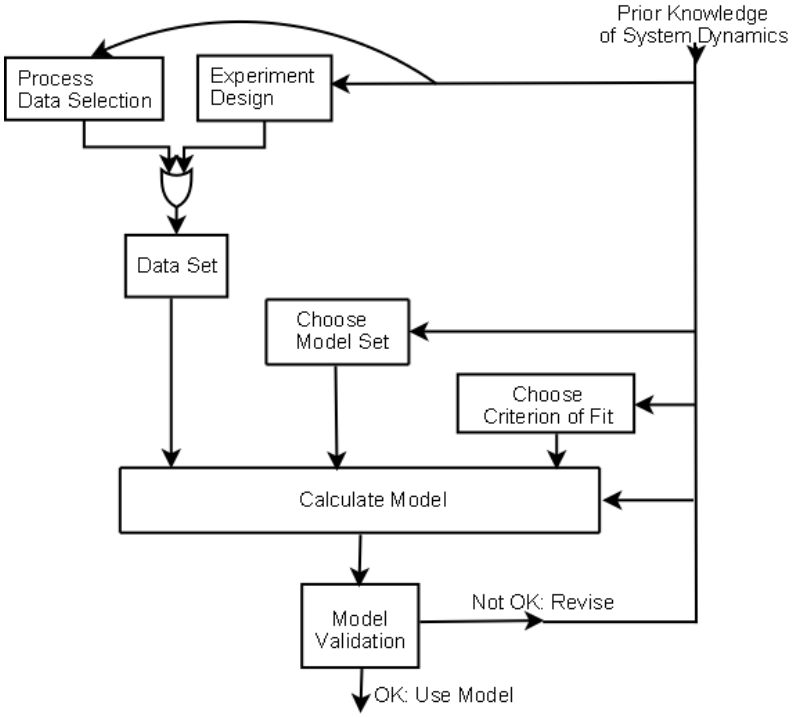

Figure 1.1: The system identification procedure. Based on [18]

1.2.3 Assessment of the Candidate Models

A model has to be selected from the set of candidate models, based on the model structures' ability to reproduce the measured data. This entity of the system identification procedure consists of choosing a criterion of fit, and calculating the model that is the best in meeting that criterion. An abstract model of the system identification procedure is shown in Fig. 1.1.

The best model from the model set can be found by estimating the parameter vector θ that minimizes a certain objective function $V_N(\theta, Z^N)$. The objective function is usually a function of the prediction errors for the time series $t = 1 : N$ of the data set Z^N, that is [18]:

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} l(\varepsilon_F(t, \theta)) \tag{1.1}$$

When a model has been chosen, it should be tested to determine whether it performs well enough - according to some criteria. It should be noted that the

resulting model from the system identification procedure is no more than the best performing model from the candidate set, given the collected data set. If, for example, erroneous assumptions have been made in the specification of the set of candidate models, or the data set gives little information with regard to the dynamics of the system, the system identification procedure is less likely to come up with a model that represents the system in a good manner. The goal of model validation is to ensure a useful model, in the sense that the model addresses the right problem and provides accurate information about the system being modelled [19].

There exists numerous parameter estimation methods. The least-squares method is a widely used one, briefly presented below.

1.2.3.1 The Least-Squares Method

The Least-Squares Method (LSE) is based on minimization of the summed square of errors between observed and simulated process output [20]. This project is constrained to linear systems, and linear regressions are very useful in describing basic linear systems. Using the concept of linear regression, a regression vector φ and the parameter vector θ, a prediction on the model output can be given as [18]:

$$\hat{y}(t \mid \theta) = \varphi^T(t)\theta \tag{1.2}$$

This gives the following expression for the prediction error:

$$\begin{aligned} \varepsilon(t, \theta) &= y(t) - \hat{y}(t) \\ &= y(t) - \varphi^T(t)\theta \end{aligned} \tag{1.3}$$

With the objective being minimizing the summed square of errors, $l(\varepsilon) = \frac{1}{2}\varepsilon^2$ becomes a natural choice. Inserting this into Eq. 1.1 gives the LSE criterion for the linear regressor given by Eq. 1.2:

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[y(t) - \varphi^T(t)\theta]^2 \tag{1.4}$$

For a parameter vector of size n, the gradient of the LSE criterion is:

$$\frac{\delta V}{\delta \theta_j} = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\frac{\delta \varepsilon(t)}{\delta \theta_j} \qquad (j = 1, 2, \ldots, n) \tag{1.5}$$

$$= -\frac{1}{N}\sum_{t=1}^{N}\varphi_j(t)\left(y(t) - \sum_{k=1}^{n}\varphi_k(t)\theta_k\right) \tag{1.6}$$

The LSE criterion is minimized when the gradient in Eq. 1.6 is equal to 0. Considering this, we rearrange the equation as in Eq. 1.7 and write it on

matrix form in Eq. 1.8.

$$\frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{n}\varphi_j(t)\varphi_k(t)\hat{\theta}_k = \frac{1}{N}\sum_{t=1}^{N}\varphi_j(t)y(t) \quad (j=1,2,\ldots,n) \tag{1.7}$$

$$(\boldsymbol{\varphi}^{\mathbf{T}}\boldsymbol{\varphi})\hat{\boldsymbol{\theta}} = \boldsymbol{\varphi}^{\mathbf{T}}\mathbf{y} \tag{1.8}$$

If the matrix $\boldsymbol{\varphi}^{\mathbf{T}}\boldsymbol{\varphi}$ is invertible, we end up with the following estimate of the parameter vector:

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\varphi}^{\mathbf{T}}\boldsymbol{\varphi})^{-1}\boldsymbol{\varphi}^{\mathbf{T}}\mathbf{y} \tag{1.9}$$

The matrix $\boldsymbol{\varphi}^{\mathbf{T}}\boldsymbol{\varphi}$ might become ill-conditioned such that the inverse does not exist. This is because the matrix contains products of the original data. Depending on the quality of the data set, this can cause the condition number of the matrix to be so high that it is deemed ill-conditioned. A better numerically behaved method for calculation of the parameter estimate is to factorize the matrix into an orthogonal and a triangular matrix. QR factorization is one such method, the theory behind which can among others be found in [4] and [18].

1.3 Identifiability and Uncertainty

This introduction to the concept of identifiability is largely based on previous work on the subject, performed by the author in [10]. The motivation for the introduction is that experience has shown that the identification of a suited model structure and a corresponding unique parameter set often is difficult. In many cases, multiple parameter sets yield equally good results in terms of a predefined objective function. This creates a problem of ambiguity in the sense that the parameter and prediction uncertainty is increased [32]. Even though the identification procedure might give a model that performs well with respect to the data set it is based on, this ambiguity reduces the certainty that the model will give a good representation of the system for input data that differs in pattern or intensity from that of the data set. If unique parameter estimates cannot be obtained, the collection of the data set and/or the choice of model structure has to be reevaluated if the value of the parameter estimates are to be meaningful [23].

Identifiability is a concept that has implications for the assessment of both the quality of the data set Z^N, and the chosen model structure $\mathcal{M}$. For the former it deals with whether the model parameters can be uniquely determined from Z^N; while for the latter it deals with whether different parameter sets can give equally performing models. The problems to be addressed in this thesis are mostly concerned with the analysis of a method to assess the identifiability of the data set, and to identify segments of process data that has propitious identifiability properties. It is however important to present definitions on identifiability with regard to model structures as well, because assumptions

based on these definitions are used extensively throughout the study. From [18] we have:

Definition 1. *A model structure $\mathcal{M}$ is **globally identifiable at** θ^* if*

$$\mathcal{M}(\theta) = \mathcal{M}(\theta^*), \quad \theta \in \mathcal{D}_{\mathcal{M}} \Rightarrow \theta = \theta^* \tag{1.10}$$

In words, this definition states that if the model structure is parametrized by θ^*, and the only possibility for having an equal model is if the parameter set of the equal model also is θ^*, then the model structure is globally identifiable at θ^*. This defines identifiability at a point, but it is naturally desireable to have a model structure that is globally identifiable for all parameter sets of the right dimension. However, it is difficult to construct such model structures [18], so a more conservative and realistically obtainable definition of 'overall' identifiability of a model structure has been introduced:

Definition 2. *A model structure $\mathcal{M}$ is **globally identifiable** if it is globally identifiable at almost all $\theta^* \in \mathcal{D}_{\mathcal{M}}$*

'Almost all' means that the model structure can be non-globally identifiable on a null set of the parameter set, i.e. on a subset of the parameter set that is negligible. If a model structure is globally identifiable, there are no two dissimilar parameter sets that give equal models.

Regarding that the core of this thesis revolves around analysis of the identifiability of a data set, it is in most cases assumed that the model structure of the systems being analyzed is globally identifiable. This is not an unrealistic assumption since the problems to be addressed are concerned with *reparametrization* of a model structure that is already in use.

1.3.1 Information Content

Let us assume that the chosen model structure is thoroughly specified based on a priori knowledge of the system at hand, and well suited to achieve a good representation of the system. More specificially let us assume that the model structure is globally identifiable. Then, the most critical part of the solution of the system identification problem is related to conditions on the data set Z^N. The data set is by far the main source of information on the system being modelled. The input signals determine which parts of the system dynamics are excited. Then, by fitting the data to the model structure through a method such as the least-squares method, a model is obtained. It is then obvious that the data set to a certain extent must exploit the dynamics of the system in order to make it possible to find a good, unambigious representation of the system.

The *information content* in the data set is a concept related to distinguishability between different parameter sets. If there are undistinguishable

parameter sets, the problem of ambiguity, as mentioned above, occures. From [18] we have the following formal definitions regarding information content:

Definition 3. *A quasi-stationary data set Z^∞ is **informative enough with respect to the model set** $\mathcal{M}^*$ if, for any two models $W_1(q)$ and $W_2(q)$ in the set,*

$$\bar{E}\left[(W_1(q) - W_2(q))\varphi(t)[u(t)\ y(t)]\right]^2 = 0 \tag{1.11}$$

implies that $W_1(e^{i\omega}) \equiv W_2(e^{i\omega})$ for almost all ω.

Definition 4. *A quasi-stationary data set Z^∞ is **informative** if it is informative enough with respect to the model set $\mathcal{L}^*$, consisting of all linear, time-invariant models.*

In a quasi-stationary signal, the expectation value and the covariance of the stochastic effects are bounded. This holds for the data sets that we are likely to encounter because the input to these systems is at least partly deterministic, while disturbances on the system are described by random variables [18]. Definition 3 tells us that a data set is informative enough wrt. to a specific model structure if having zero (summed) variance of the difference between two models outputs requires that the models are equal. Definition 4 is a specification that gives conditions on distinguishability between models in linear systems.

Considering that calculation of the variance is a way to capture the model output differences' degree of being spread out, the definition tells us that, as we stated above, a data set is informative enough with regard to a specific model set if: for two models to give equal performance, the models, i.e. the parameter sets, have to be equal. Thus, there will only be one unique parameter set that gives the best representation of the system if the data set is informative enough. This implies identifiability of the system, given a well designed model structure.

The information content of a data set is closely related to a concept regarding the input data; *persistence of excitation*.

1.3.2 Persistence of Excitation

The underlying assumption of the system identification procedure for linear systems is that the measurements $y(t)$ are linear with respect to both the unknown parameters θ and the states of the system $x(t)$. Since the system states, and thus the output, can only be manipulated via the input signal $u(t)$, we need conditions that ensure an informative data set [21]. *Persistence of excitation* is a concept that give such conditions.

The following is based on the discussion on persistence of excitation in [18]. A formal definition is presented:

Definition 5. *A quasi-stationary signal $u(t)$, with spectrum $\Phi_u(\omega)$, is said to be* ***persistently exciting of order n*** *if, for all filters of the form*

$$M_n(q) = m_1 q^{-1} + \cdots + m_n q^{-n} \tag{1.12}$$

the relation

$$|M_n(e^{i\omega})|^2 \Phi_u(\omega) \equiv 0 \text{ implies that } M_n(e^{i\omega}) \equiv 0 \tag{1.13}$$

For parameter convergence it is required that the input signal is persistently exciting of at least the same order as the total number of parameters in the model structure. Usually, there are $2n$ parameters in an n-th order linear system, implying that the input must be persistently exciting of order $2n$ for such a system. The effect of Definition 5 is that the input is *persistently exciting of order n* if the power spectrum of the input signal has at least n non-zero fundamental frequency components. A stronger version of the concepts is that a signal is said to be *persistently exciting* if the spectrum $\Phi_u(\omega)$ is non-zero for almost all ω.

If you perform an open-loop experiment on the system to collect the data set, this experiment would be informative (see Def. 4) if the input is persistently exciting. In practice, this implies that the input signal must contain at least as many distinct frequency components as the number of parameters to be estimated. It is not very complicated to design an experiment with an input signal that meet this requirement. However, when you have to use process data from the normal operation, without any influence on the input signal, matters get more complicated with respect to finding a data set that is persistent exciting of a sufficient order.

For a closed loop system, the basic convergence theorem applies, i.e.: a prediction error assessment method (e.g. least-squares) will estimate the system if the data set is informative and the model set contains the true system. This fact is important because it let us use a similar approach to find an informative data set from a large amount of process data both for open and closed loop systems. However, it must be mentioned that a requirement for obtaining an informative data set from a closed loop system is that the *setpoint* of the controlled system is persistently exciting [18]. For a practical setpoint-regulated system, this implies that to find an informative data set, one will have to identify process data segments where there are considerable setpoint changes. A lot of control systems, this includes many petroleum production control systems, has control loops where the setpoints are constant for long periods of time. An example could be the control of an oil well where it is important to maintain a desired well-head pressure by adjusting the flow rate of the fluid being injected back to the well [22]. If, for some reason, a reparametrization of the model that is used in the control system is wanted, without the ability to stop the control system to perform open loop experiments; one will have to search through the process data for segments where

the system seems to be going through a change of operation mode in order to find periods of informative data.

1.3.3 Uncertainty Analysis

Generally speaking, uncertainty reflects the lack of sureness about something. In a more specific context, uncertainty can be divided into two subconcepts; natural variability and epistemic uncertainty, i.e. uncertainty due to lack of knowledge. Natural uncertainty is a property of the system, it refers to the randomness of the nature of the system. Knowledge uncertainty reflects how well you know the nature of the system. Analysis of the knowledge uncertainty can, in theory, be used to reduce the overall uncertainty of the system. Parameter uncertainty describes the uncertainty of using a parameter set based on incomplete system knowledge or a data set with a low information content. [26]

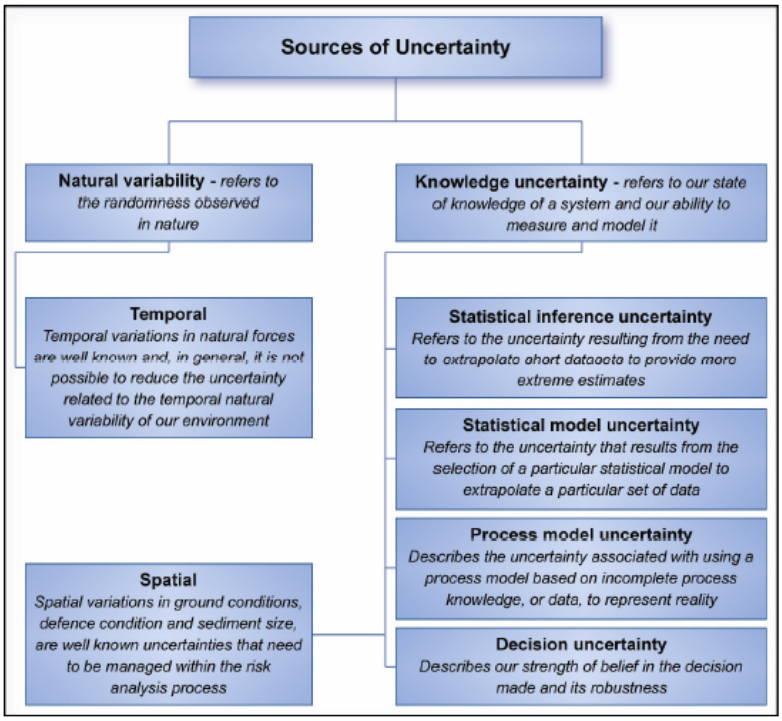

Figure 1.2: Sources of uncertainty. (Sayers et. al. 2002 [26])

Uncertainty and sensitivity are two notions that can cause a bit of confusion with respect to an analysis of how parameter value changes affect the model output. One misconception, or at least an exaggerated simplification, is that sensitivity is related to how variations in the output are based on vari-

ations of the input and the model parameters, whereas uncertainty merely concentrates on determining the lack of certainty in the model output performance. However, many uncertainty estimation methods also deal with parameter identifiability, ambiguity and uniqueness. This has caused sensitivity analysis to become an integrated part of such uncertainty analysis methods [12].

The goal of an uncertainty analysis is to quantify the overall uncertainty of the model as a result of uncertainties in the sources listed in Fig. 1.2. We will here foucs on the analysis of process model uncertainties. Since a general assumption used for the system identification/parameter estimation problem in this thesis is that the model structure is globally identifiable and that the set of candidate models include the structure of the true system, we can further specify the focus of the uncertainty analysis to the parameter uncertainties.

The idea of an uncertainty analysis with respect to assessment of parameter uncertainty, is to observe the behaviour of the model output when uncertainty is injected into the model. One way of doing this is by selecting the parameter values from a probability distribution of a specified feasible parameter space. Then, the model performance, with respect to a certain objective function, is observed for numerous, semi-randomly chosen parameter sets. From this we get a measure on the parameter uncertainty, i.e. the uniqueness of the parameter set(s) that give the best model performance. This can be called a sample-based parameter uncertainty analysis. Reasonable upper and lower limits for the different parameter values, and a thought-through selection of the probability distribution function (PDF) and the number of parameter sets to use in the analysis is important in order to reduce computational complexity while maintaining good model assessment properties. These choices should be based on a priori knowledge of the system, and made independently from case to case. The implementation of an uncertainty analysis method is often a trade-off between the quality of the results and the computational difficulty of the analysis.

Principles for good practice of the conduct of a sampling-based uncertainty analysis as described above has been given by [24]:

- When deciding upon a PDF for parameter values, consider the following questions:
  (a) Is there any mechanistic basis for choosing a distributional family?
  (b) Is the PDF likely to be dictated by physical, geological or other properties and mechanisms?
  (c) What are the bounds on the parameters?
  (d) Is the PDF symmetric or skewed, and if skewed, in which direction?
- Base the PDF on empirical, representative data

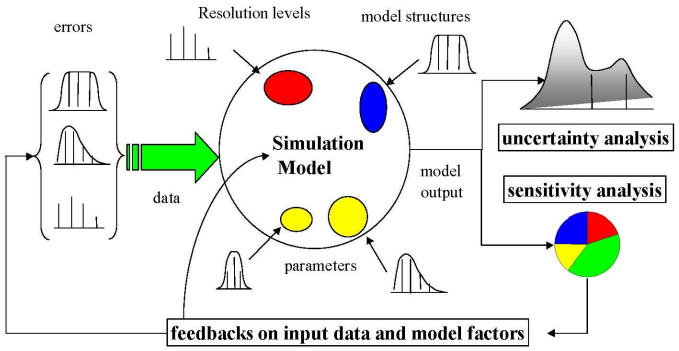

Figure 1.3: Scheme of a sampling-based sensitivity / uncertainty analysis. Courtesy of Andrea Saltelli

In the traditional literature on system identification there seems to be little to no discussion on the use of uncertainty analysis. There is agreement in that uncertainty is an important factor in oil and gas production optimization [3]. However, explicit treatment of uncertainty in production optimization with system identification has received little attention, until the recent doctoral thesis of Steinar M. Elgsæter from NTNU [7].

The relationship between uncertainty analysis and information content in the data set and identifiability may seem obscure at the time, but as the deduction of the suggested method for parameter estimation from large amounts of process data is brought about, the relationship should become clear. In short, it can be said that high uncertainty of a parameter value can be interpreted as a result of lack of information content in the data set. This can later be thought useable to generate an estimate of the information content for different data segments over the time series that is being analysed.

1.4 Outline of the Thesis

The structure of the thesis is organized as follows:

Chapter 2 is devoted to a thorough presentation of the DYNIA method and a discussion on the challenges of system identification with respect to uncertainty.

Chapter 3 presents and documents an implemenation of DYNIA. The implementation is tested by performing two examples and a case study.

Chapter 4 discusses the industrial applicability of DYNIA. Pros and cons for the practical use of the method are considered.

Chapter 5 presents a number of modifications that may have the capacity to improve the functionality of DYNIA.

Chapter 6 contains final conclusions and remarks, and suggestions for further work and research.

2 Dynamic Identifiability Analysis - DYNIA

In [10] it was discussed how it is likely that performing an uncertainty analysis and the calculation of a likelihood distribution for each parameter will uncover potential lack of parameter identifiability and lack of information content in the data set. A likelihood function can be defined as

$$\theta \rightarrow f(y = y_m | \theta = \theta^*), \tag{2.1}$$

which is a conditional probability distribution function for the system output that give the probability of the model output y_m being equal to the system output y, given that the model uses the parameter set θ^*. It was shown in [10], with an example of how the maximum a posteriori (MAP) parameter estimation method (see Appendix6) works, how performing a sample-based uncertainty analysis as described in the previous section give us all the information needed to be able to calculate a likelihood distribution for θ. A sample-based uncertainty analysis test the performance of the model with a range of parameter sets and, just as the MAP parameter estimation method, can use this to obtain a distribution of the parameter value likelihood. Upper and lower confidence limits for the value of each parameter value can be calculated based on these likelihood distributions. As mentioned in [10], a logical detection is that if a parameter has a narrow confidence band, this is a sign of low uncertainty, high sensitivity to parameter value changes with regard to model performance, and high information content in the data set with respect to this parameter (when applying the assumption that the model structure is globally identifiable), and an intuitive hypothesis is that uncertainty analysis can be employed in the search for an informative data set. By determining how a model's uncertainty and likelihood towards different parameters vary over time in a data set, the hypothesis is that one can use the width of the likelihood confidence band as a measure of how the information content of the data set, with respect to these different parameters, vary with time. Thus, one could identify segments of the process data where the information content is high, and use these segments in the system identification procedure to obtain, potentially, a unique set of parameter estimates [10].

The search for segments of a data set with a high information content has only to a minor extent been discussed in the traditional system identification literature. Ljung [18] dedicate one paragraph to the subject. He states that when there are long periods of missing data, considerable disturbances or data that seem to contain 'no information', it is natural to select segments of the original data set which are considered to contain relevant information about dynamics of interest. However, it is also stated that the procedure of how to select such segments '*will basically be subjective and will have to rely mostly upon intuition and process insights* [18], [10]. As was mentioned in the previous chapter, the recent doctoral thesis of Steinar M. Elgsæter [7] suggests an approach to model-updating for the offshore production of oil and gas that is based on parameter estimation against production data stemming from normal operation, taking into consideration the challenges that arise when there is low information content in the data set. Elgsæter mentions that this suggested approach is motivated by a desire for an updating scheme '*which requires little human intervention and does not require frequent additional experimentation, for instance in the form of well tests.*' In the next section system identification under uncertainty is discussed, with a connection to the work of Elgsæter. This is followed by the presentation and review of dynamic identifiability analysis (DYNIA), which is the author's suggested method to identify segments of the data set with high information content.

2.1 System Identification under Uncertainty

The basic problem of the system identification procedure is to estimate unknown model parameters for a given system from observed data. To achieve uniqueness in the parameter estimates, the importance of which has already been discussed, there needs to be sufficient information content in the data set, i.e. the data set has to be informative enough with regard to the selected model structure. If the information content in the data set is low, the system identification procedure is prone to find multiple parameter sets that give equal performance. Thus, the parameter uncertainty increases when the information content is low. Classical system identification literature considers the design of experiments to obtain a data set with high information content without considering the cost of performing experiments. The associated costs of these experiments may make this approach to reduce or eliminate parameter uncertainty impractical, and it is preferable to fit system models to production data stemming from normal operation [7]. This section discusses challenges that arise when the system idenfication procedure is performed with sources of uncertainty present, and suggests steps that should be taken to improve system identification performance under such circumstances.

There are three main reasons why a system model might not describe the system accurately; model structural uncertainty, measurement uncertainty

and low information content in the data set [7]. Model structural uncertainty means that the model is a member of a model structure that is unable to describe the dynamics of the system, regardless of parameter values. In these cases the model structure is not globally identifiable (see Eq. 2). As mentioned in Section 1.3, it is for most of this thesis assumed that the selected model structures are indeed globally identifiable. Consequently, while model structural uncertainty is a matter that should not be taken lightly, it is beyond the scope of this text to elaborate on it. Model structural uncertainty is also introduced when the user has reason to believe that parameters or model structure will change over time. This has to be evaluated on a case-to-case basis, taking into consideration the changes that the user foresee.

Measurement uncertainty is uncertainty about the difference between measured and actual input, state and output values. It is important to recognize that all measurements are wrong in that the measured and actual value are different. The difference between these two is the measurement error. A statement of measurement uncertainty indicates how large, given a certain level of confidence, the measurement error might be [1]. The higher the measurement uncertainty, the more it deteriorates the results of the system identification procedure, degrading model performance. Incorrect calibration of measurement equipment and unsatisfactory resolution of the measurement values are two important sources for measurement uncertainty. A more detailed discussion on measurement uncertainty is well beyond the scope of this text, and will not be done here.

The last main source for degraded model performance is low information content in the data set. Information content in process data is related to the persistence of excitation of the input signal. The connection between information content and parameter uncertainty has been discussed in sections 1.3.2 and 1.3. Elgsæter [7] presents an extensive case study for a typical offshore oil and gas production field. In this study the production data from normal operation is assessed with regards to modeling for production optimization. It is stated that the difficulties posed by the production data in a context of information content and measurement uncertainty are not unique to this field, and it was attempted to make the study as independent to the choice of production model as possible. The conclusion of the case study is clear: the production data was observed to have low information content and be subject to significant measurement uncertainties. Elgsæter highlights three research challenges which should be faced to improve system identification performance under such circumstances, but that has received little attention in the literature; firstly, uncertainty in the production system should be estimated to assess its significance with regards to lost production potential; secondly, proposed actions to reduce uncertainty should be evaluated by estimating costs and values of such actions; thirdly, strategies for making decisions in day-to-day operations under uncertainty should be investigated.

It has been mentioned in this text that using process data from normal

operation as opposed to data obtained by designed experiments is desirable because it reduces the cost of the process that has to be done to recalibrate model parameters. It is however important to solve the problems that arise when using a data set that has low information content before embarking on this route, because fitting models to a data set that has low information content will increase parameter uncertainty significantly. Elgsæter [7] mentions that significant parameter uncertainty increases the probability that the system identification procedure will suggest a parameter set that is erroneously assumed to describe the system, and can cause the control system to suggest setpoint changes that are sub-optimal and may reduce profit. He proposes that rather than giving in to these problems and abandon the use of production data from normal operation, parameter uncertainty should be quantified or estimated and he suggests that further work on the subject should focus on '*exploiting this quanitifcation of uncertainty to devise strategies for production optimization under uncertainty.*' This is very much in line with the main contribution of this thesis.

In Section 1.3, and earlier by the author in [10], it was mentioned how a sampling-based uncertainty analysis give a measure on the parameter uncertainty. Elgsæter [7] suggests using bootstrap resampling to numerically estimate parameter uncertainty. Bootstrapping works by estimating the process that generated the data by an approximating distribution from which samples may be drawn. N_s bootstrap data sets are obtained and sampled from this distribution, and system identification is used to determine a parameter set, $\hat{\theta}$, for each bootstrap data set. The distribution of $\hat{\theta}_i \forall i = 1, \ldots, N_s$ is an estimate of the parameter uncertainty [7], [16]. Even though bootstrap resampling is a straightforward way to estimate parameter uncertainty it has certain properties that makes it a poor choice for the DYNIA method, or one could argue the opposite; that DYNIA has certain properties that makes bootstrap resampling a poor choice. This will come forth in the presentation of the DYNIA method.

2.2 The DYNIA Method

Dynamic Identifiability Analysis (DYNIA) is an attempt to avoid the loss of information through aggregation of the model residuals in time. The method can be used to estimate the amount of information available to identify a specific parameter or to detect failure of model structures in an objective manner [32]. The methodology was developed by Dr. Wagener, assistant professor of civil engineering at Penn State University, and first presented in an article in 2002 [31], and in more detail in another article one year later [32]. These are the only articles found that describe how the method works. The articles are written in a practical user-friendly manner, and a mathematical review of the basis on the fundamental functionality of the method has not

yet been published. DYNIA has its origin in the field of hydrology, but it does not make any assumptions that should degrade the performance of the method with regards to any other general dynamic mathematical model.

Wagener has implemented a version of the DYNIA method that is available in a Matlab toolbox called MCAT [33]. The source code of the method is protected and not available to the user. To get a deeper understanding of how DYNIA works and to be able to suggest and test modifications to it as well as providing a basis for further research, it was deemed important to come up with a new implementation of the method. This implementation will be documented later in this chapter. Before that though, a general presentation of the method is appropriate.

DYNIA is an approach to improve the amount of information that can be retrieved from observations for model evaluation. The methodology is based on elements from Regional Sensitivity Analysis (RSA) ([27], [13]) and the Generalized Likelihood Uncertainty Estimation Framework (GLUE) ([11], [29], [2]).

The basis of RSA is an investigation of whether an initially known parameter probability distribution function(PDF) changes when it is conditioned on a measure of performance, in this case an objective function. Parameter sets are typically sampled from an independent uniform PDF for each parameter. Model performance is evaluated for each parameter set with regard to an objective function such as the sum of squared errors (SSE), and the parameter sets are ranked according to model performance. A post simulation PDF is calculated for each parameter, conditioned on model performance. Differences between the initial and the conditioned parameter distribution indicate not only parts of the distribution that performs well and poorly, but also the sensitivity of the model response to changes in the parameter [32], [27].

GLUE is an uncertainty estimation technique based on Monte Carlo simulation where the model output for each parameter set is evaluated with regards to a goodness-of-fit criterion, typically chosen as *SSE*. Models with parameter sets that give an acceptable goodness-of-fit, according to a certain acceptance limit, are retained in a behavioural set, while parameter sets whos model performance are below the acceptance limit are rejected from the rest of the analysis. The goodness-of-fit, in this case the SSE values, for the parameter sets in the behavioural set is used to construct a likelihood measure of each parameter set. For each point in time, the parameter samples of the set of behavioural models are weighted according to a likelihood weight, associated with each behavioural parameter set, in forming a cumulative probability distribution function of the model output value at that point in time [2] [33] [29]. Uncertainty intervals for the model output are obtained based on the calculation of confidence limits of this distribution. The likelihood measure associated with each parameter shall be positive and increase monotonically with increasing model performance, and the method allows updating of likelihood weights as new data becomes available for model evaluation. Figure 2.1

shows an example of the output when using the GLUE method to estimate output uncertainty. Observed values are plotted together with the associated upper and lower confidence limits. The bottom plot on the figure shows the width of the confidence interval at each time step, making the identification of regions with large uncertainties easier.

A disadvantage of the GLUE method is the subjective choice of likelihood function and the assessment of an acceptance limit that separates behavioural and non-behavioural models, i.e. how bad the performance can be before it can be rejected as to have no probability to represent the system [33]. However, the Monte Carlo sampling procedure enables the user to tune the uncertainty estimation through the choice of objective function and a priori knowledge combined with experience-based decisions regarding the expected performance of a model in the face of for example data errors. The performance of GLUE should also be tested by verification of the results on independent data sets.

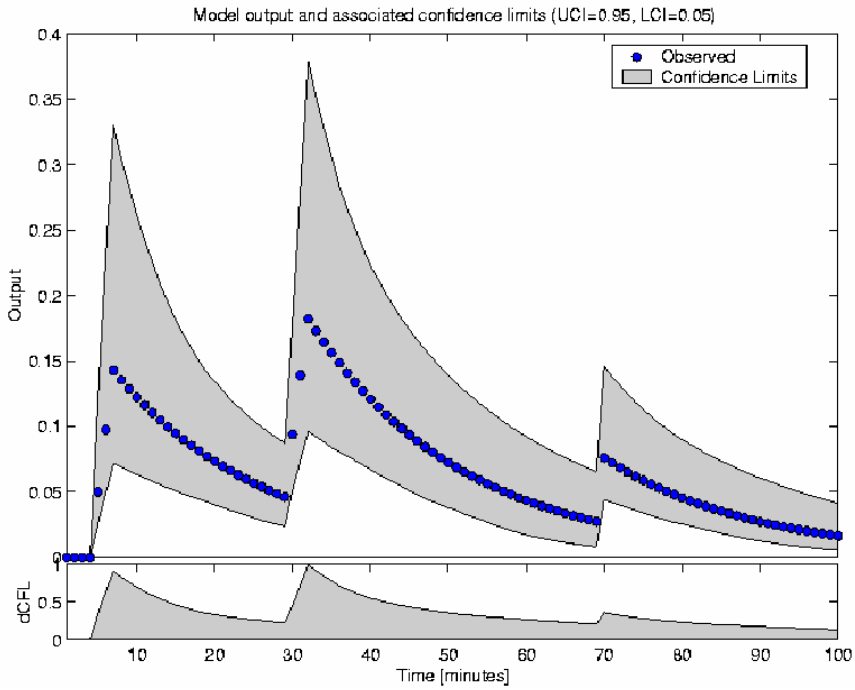

Figure 2.1: GLUE output uncertainty limits (from [33])

As mentioned,the DYNIA method draws on elements from both RSA and GLUE. The basics steps of DYNIA are shown in Figure 2.2. These steps has earlier been presented by the author in [10]. The describtion of them herein goes more into details, but is also influenced by the earlier presentation.

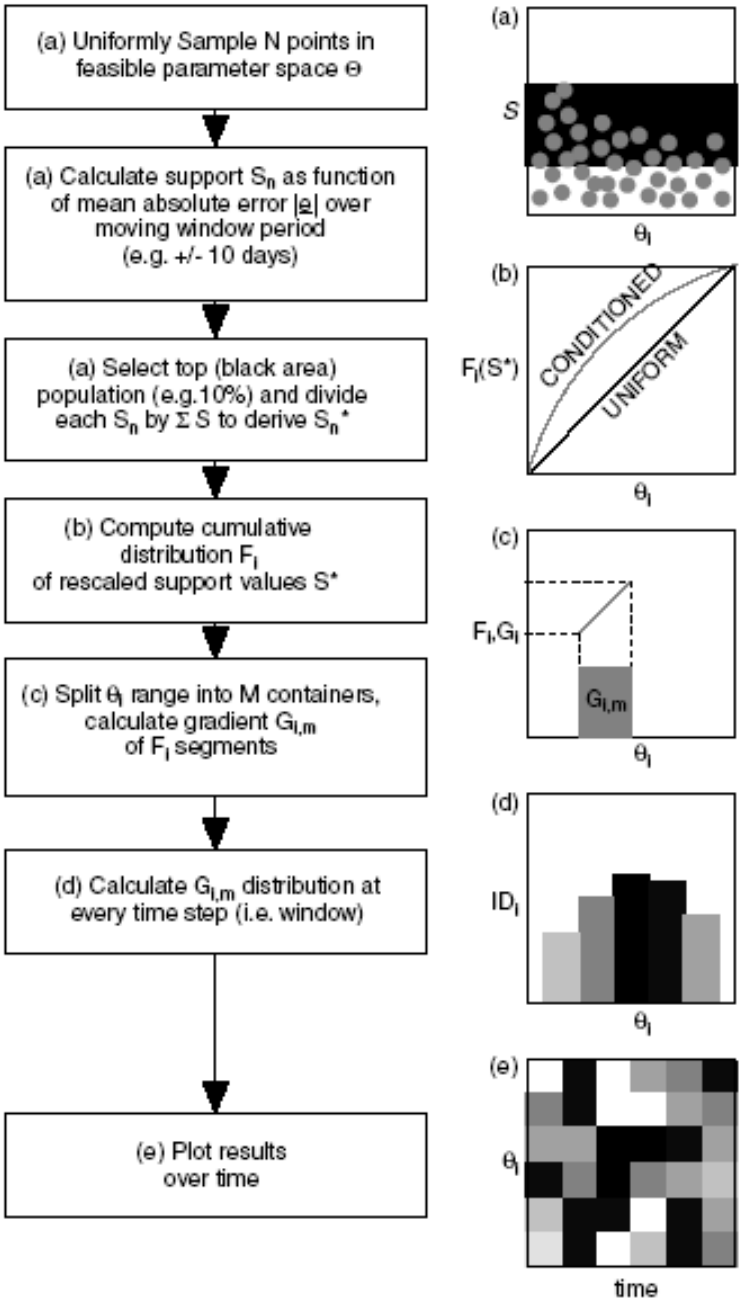

Figure 2.2: The DYNIA procedure (Wagener et. al. 2003 [32])

2.2.1 Monte Carlo Parameter Sampling

Monte Carlo parameter sampling is a computational algorithm that repetitively perform model simulations with parameters randomly sampled from an a priori probability distribution. It is assumed here that a globally identifiable model structure for the system is known beforehand, so that it is known beforehand what parameters are to be analysed. With the DYNIA method, each parameter is individually and uniformly sampled within an interval of feasible parameter values. The main area of use for this method may be reparametrization of a model. This imply that earlier used parameter values combined with an insight of the dynamics of the system could be used to suggest feasible parameter values. This is particularily true for white-box and grey-box models, where most or all of the parameters have a physical meaning.

Simulated model output and corresponding objective function value, for example SSE, is stored for each sampled parameter set for later assessment. Figure 2.3 shows an example of the SSE values corresponding to one specific parameter of Monte Carlo sampled parameter sets. This plot and the rest of the plots in this section are included to clarify the concepts that are described in the text. For the curious the plots stem from the analysis of a linear storage model with two parameters; gain and residence time (rt).

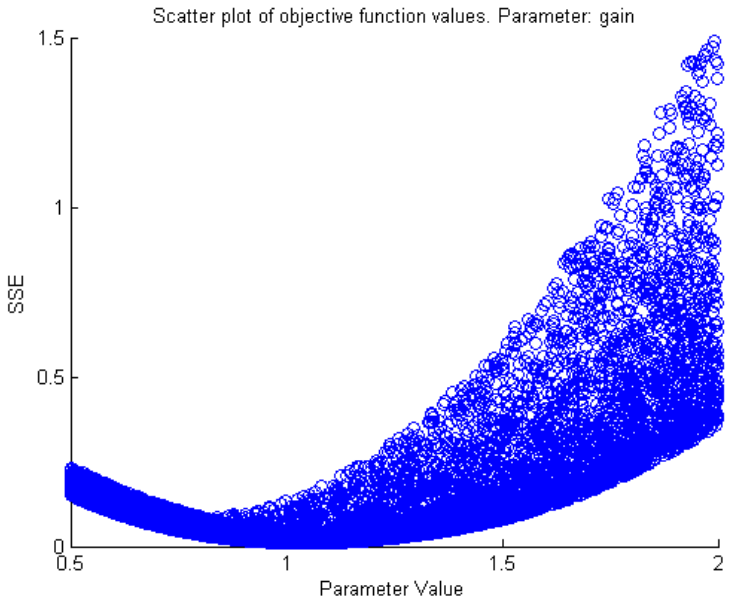

Figure 2.3: Example plot of SSE values corresponding to one specific parameter of Monte Carlo sampled parameter sets

Monte Carlo sampling can often be computationally heavy. To achieve

good repeatability and accuracy of the results from the sampling and simulation procedure it is important to have a good coverage of the feasible ranges for each parameter. Consequentially it is in most situations necessarily to perform a large amount of samples.

It has been stated that Monte Carlo sampling using a uniform prior PDF over a relatively large parameter space, can result in an algorithm that after billions of model evaluations may not have generated even one good solution [29]. If numerous parameters are to be analysed, and there is a lack of a priori system knowledge in order to come up with a compact parameter space, the computational requirements are likely to become very large. The obvious problem is that the basic Monte Carlo sampling technique suffers badly from *the curse of dimensionality*, i.e. the requirement that the number of samples per parameter variable increase exponentially with the total number of parameter variables in order to maintain a given level of accuracy.

When performing the sampling procedure, the principles for sample-based uncertainty analysis given in Section [**?**] should be employed. Some other suggestions on how to reduce the computational demands of the Monte Carlo sampling procedure follows [10]:

- One obvious suggestion is to increase computing power.
- Perform a preliminary parameter estimation using non-sampling based methods over a subjectively selected part or all of the available time series data. Depending on the result, the information gathered can be used to reduce the feasible parameter space and/or to find a non-uniform a priori parameter PDF.
- Perform the DYNIA method on pre-selected data segments. The selection of data segments to study will have to rely on intuition and process insights.
- Applying a *stratified sampling* method, such as *Latin hypercube sampling*. Given n parameters with upper and lower bounds, each parameter can be divided in m equally probable intervals. This will give a *hypercube* consisting of M divisions and N parameters giving C possible combinations, where

$$C = \prod_{n=0}^{N} (M-n)^{N-1}, \tag{2.2}$$

  with the requirement that $M > N$. A few samples are made within each component, until all components have been sampled. This give better control of how many samples you need to achieve a certain accuracy. It should also be possible to combine it with a non-uniformal PDF. Then, the number of components could be reduced and more samples made in components where one or more parameters have a certain probability.

Later in the thesis, modifications to reduce the complexity of the DYNIA method are discussed in more detail.

2.2.2 Constructing a Measure of Likelihood and Identifiability

The second step of the DYNIA method introduces elements from the GLUE methodology. First, the parameter sets from the Monte Carlo sampling are ranked based on their SSE value. As with the GLUE technique, the parameter sets are deemed behavioural or non-behavioural based on the model performance they give. Only the behavioural parameter sets are considered further on in the analysis. In the DYNIA method, a collection of the parameter sets that give the least SSE value are deemed behavioural. [32] suggests the top 10% performing parameter sets to be used further on, but this is a consideration where user experience and knowledge of the system at hand has to be influential. The behavioural set for the parameter shown in Figure 2.3 is shown in Figure 2.4.

It has been suggested to use more sophisticated methods of specifying an acceptance threshold. An example is to accept models that describe certain characteristics of the response well by solving a multi-objective optimization problem [33]. This is not treated in greater detatil here.

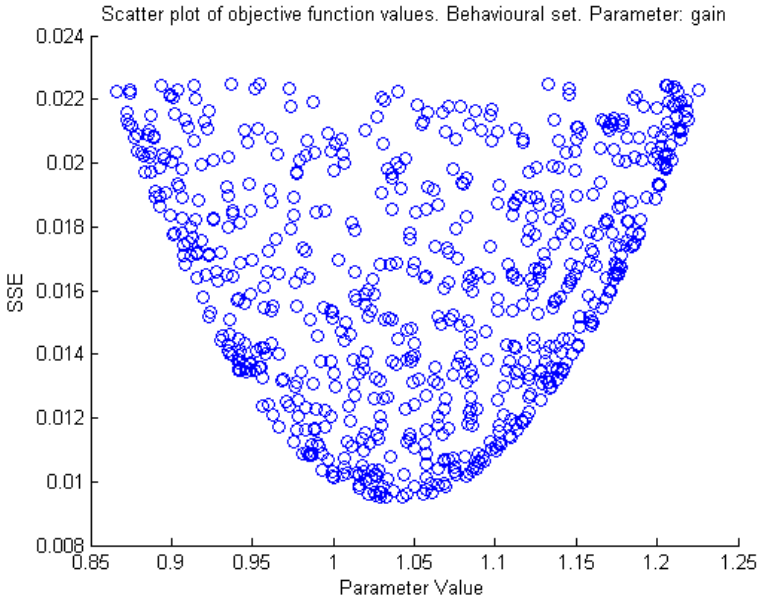

Figure 2.4: SSE values of the behavioural set of Fig. 2.3

Another element from the GLUE method is the construction of a likeli-

hood measure. In both GLUE and DYNIA, likelihoods are regarded as any performance measure that can be used to differentiate how likely it is that the model is representative for the system at hand. The main requirements are that the likelihood measure is monotonically increasing, above zero and that it adds up to one. The MCAT toolbox construct the likelihood measure for each parameter set with regards to the objective value (usually SSE) of the model simulation for each parameter set. The following pseudo code describes this [33]:

```
likelihood = 1 - objective_value;

if min(likelihood) < 0
    likelihood = likelihood - min(likelihood);
    end

likelihood = likelihood / sum(likelihood)
```

This is obviously a very simple likelihood measure to calculate, but it clearly meets the mentioned requirements. However, this method has been criticised for treating likelihood in a wider context than the traditional statistical one [29]. It is stated that it is critical that the likelihood measure function depend on the determination coefficient, R^2, of the maximum likelihood estimate, and that Bayesian statistics should be used to construct it. Section **??** give some insight on the use of Bayesian statistics to calculate a likelihood function. Nevertheless, the described likelihood measure is being suggested used by the developer of DYNIA, and it will also used here. Possible modifications are presented later.

Whereas the GLUE method uses the likelihood measures to estimate model output uncertainty, the DYNIA method at this point draws use of elements from the RSA method in order to change the focus over to the estimation of parameter uncertainties. The likelihood measures are used to create a cumulative distribution of the likelihood (CLD) for each parameter. Assessment is then performed based on the shape of the CLD. Figure 2.2.b is an example of this. When this plot show a (nearly) straight diagonal line, it implies that the likelihood is about equal over the parts of the parameter space that are members of the set of behavioural parameter values. This is evidential of a poorly identified parameter, low sensitivity of the model response to changes in the parameter value, and high parameter uncertainty as there is a low degree of uniqueness of the most optimal parameter values. An estimate on the parameter uncertainty is easily obtained by calculating upper and lower uncertainty limits corresponding to a specific uncertainty interval (e.g. 95%).

In the opposite case, when the plot of the CLD is curved, i.e. conditioned on the likelihood weighting, which in turn is conditioned on model perfor-

mance, you have evidence for a better identified parameter. The narrower the CLD is, i.e. there is a small behavioural set, and when it is steep over a small part of the feasible and behavioural parameter values, you have indication on a strongly identified parameter. The close relationship between the identifiability of a parameter, the sensitivity of a model to parameter changes and the uncertainty of the 'true' parameter value has been discussed earlier in the thesis. The connection between information content in the data set and these concepts has also been discussed. DYNIA exploits these connections to interpret the conditioning of the CLD in the context of information content in the data set. This will be described in the next section.

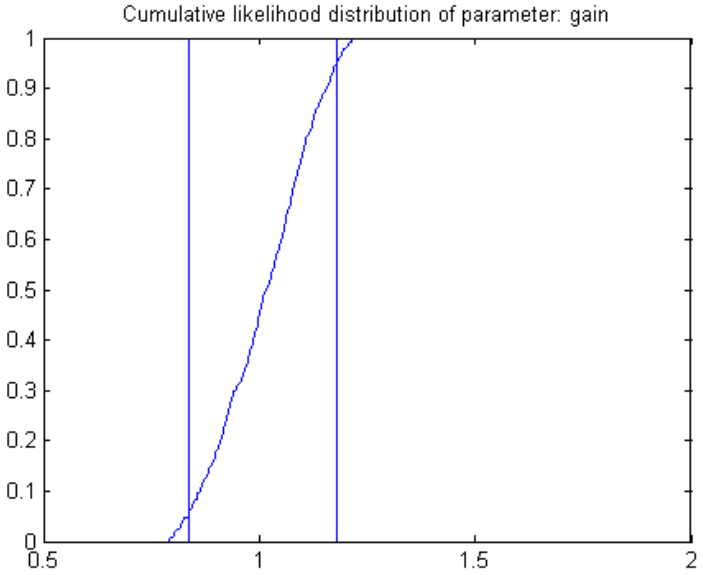

Figure 2.5: A parameter with a low uncertainty band and steep CLD, indicating high information content in the data set wrt. this parameter. The same parameter as in Fig. 2.3

Figures 2.5 and 2.6 shows examples of the plot of CLD and uncertainty bands. For both parameters the Monte Carlo sampling had a feasible parameter space equal to the entire x-axis of each parameter.

Further on in the DYNIA method, the parameter range is divided into an arbitrary number of regions. A high number of regions increase computational complexity later, but can give more accurate results. 20 regions is suggested as an example [33]. The gradient of the CLD is calculated in each region. These gradients will indicate the average strength of conditioning to the objective function for the parameter values within that specific range. In the DYNIA method, this is looked upon as an indicator of the identifiability of each range,

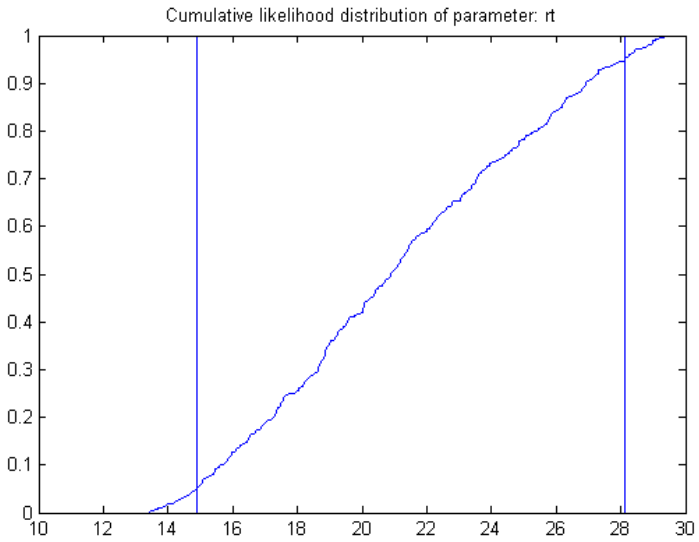

Figure 2.6: A parameter with a wide uncertainty band and CLD with a gentle slope, indicating low information content in the data set wrt. this parameter

resulting in that the region with the highest gradient marks the location of greatest identifiability of the parameter [32]. Figure 2.7 shows an example plot.

2.2.3 Moving Window Calculation of Likelihood and Identifiability Measures

The third and final main step is where the likelihood and identifiability measures are employed in a dynamical manner to arrive at a method that give a measure on the information content and the identifiability of the process data *as a function of the time series*.

Dynamic employment of likelihood and identifiability here means that a cumulative likelihood distribution (CLD) is calculated at each time step. This is done in a moving-window approach. At each time step t in a data set with a constant sampling interval, a window is created consisting of t and the n time steps before and after t, for a total of $2n + 1$ time steps. $2n + 1$ is called the window size. Each time step has its own appurtenant window, and the methods described in the previous section are performed for each window. Thus, for each window a behavioural set is obtained from the Monte Carlo sampled parameter sets, and likelihoods, CLD and gradients of the CLD are calculated. This give an aggregated CLD at each time step. Notice that the Monte Carlo sampling is only performed once.

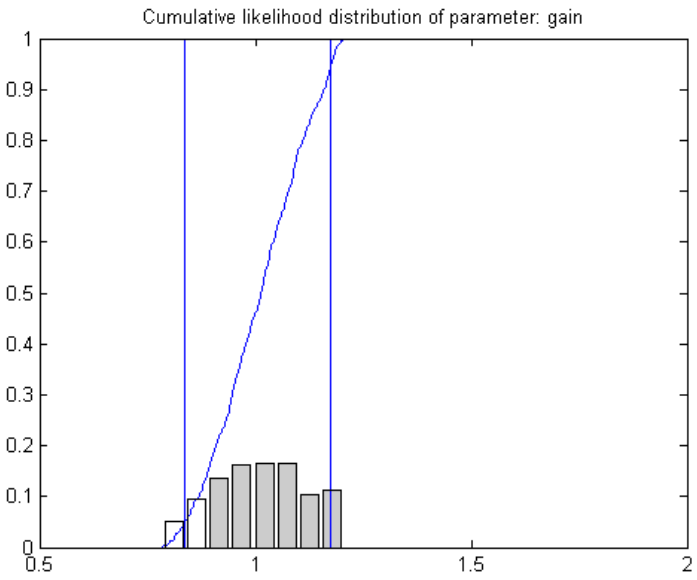

Figure 2.7: Plot of the CLD and gradients (eight ranges) for the same parameter as in Fig. 2.3

The window size depends upon the length of the period where the parameter is influential on the system dynamics, for example after a setpoint step change. This will in most cases to some degree have to be subjectively chosen or . A large window size is less prone to e.g. measurement errors, but is not suited for systems where the parameters have short influential periods.

2.2.3.1 Information Content

As mentioned, the parameter uncertainty is estimated by calculating upper and lower uncertainty limits from the cumulative likelihood distribution (CLD) for each window. In the DYNIA method the width of this confidence band is not only regarded as an estimate on the parameter uncertainty, but with the dynamic employment of the results provided by the moving-window approach also as a measure on the information content at and around the midpoint of each window. A narrow confidence band in the CLD of a window implies that there is high information content because the uncertainty is low and the sensitivity to parameter changes high.

In the MCAT toolbox, a 90% confidence interval is suggested as a versatile, 'allround' value. It is however hard to give firm advices on this because it depends on the quality of the data set.

A plot that shows how the information content for a parameter vary with

the time series of the process data can then be generated. An example is shown in Figur 2.8. The value of the information content (IC) at each time step t is suggested to be calculated as:

$$IC(t) = 1 - \frac{p_u(t) - p_l(t)}{p_{max} - p_{min}}, \tag{2.3}$$

where p_u and p_l are the parameter values at the upper and lower confidence limits, and p_{max} and p_{min} are the maximum and minimum acceptable values of the parameters in the behavioural set. This normalizes the estimate of the information content to a value between 0 and 1, where a higher value means more information content.

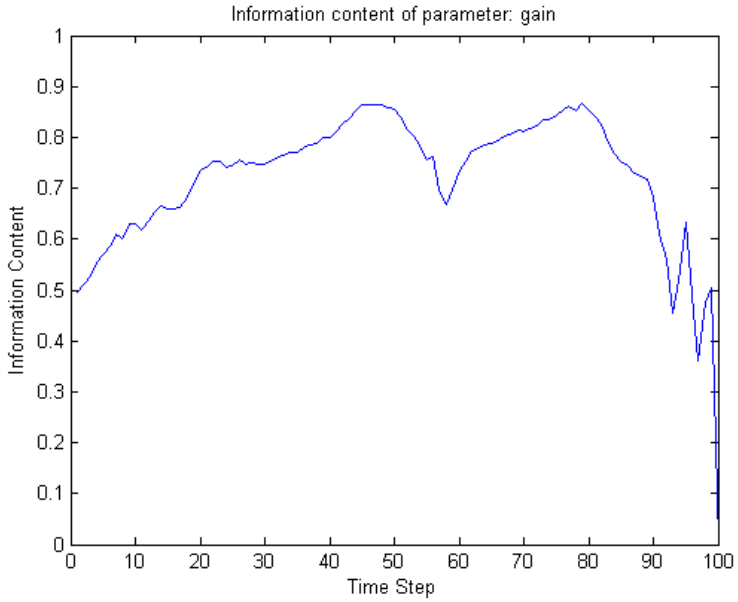

Figure 2.8: Plot of the information content of the same parameter as in Fig. 2.3

2.2.3.2 Parameter Estimation with DYNIA

The dynamic employment of the principles of DYNIA can be taken one step further to get a complete parameter estimation method. The user may only be interested in identifying data segments with high information content and estimate the model parameters with a traditional method, e.g. the least-squares method described in 1.2.3.1, but parameter estimation with DYNIA is nevertheless presented here.

The basis of the method lies in combining the cumulative likelihood distribution (CLD) in each window with its gradien. First, the gradient is used to

obtain an aggregated identifiability distribution within the uncertainty band of the CLD for each window. The upper and lower bound of the uncertainty band indicate the range that the 'true' parameter value is a part of. The information content is a normalized version of the uncertainty band, better suited to compare data segments for different parameters and models. The identifiability distribution, on the other hand, indicate the parameter values from the uncertainty band that give the best model performance. If the CLD has a gradient that is high for a small range of parameter values, but low for the rest of the uncertainty interval, it indicates a high degree of identifiability and uniqueness for the parameter values in that range.

Next in the parameter estimation, the uncertainty band of the CLD for each window is plotted, along with its gradient, over the time series. The gradient at each time step is indicated by varying the color inside the uncertainty band, depending on gradient value. The difference of this plot compared to the plot of the information content is that instead of showing how the information content vary with time, with a value between 0 and 1, it shows the uncertainty band, with parameter values, at each time step, and in addition indicates the identifiability of the parameter values inside that band. The optimal parameter value estimates are then likely to lay in an area of the plot where the uncertainty band is narrow and the gradient of the CLD reaches its highest value. Figure 2.9 shows an example of this.

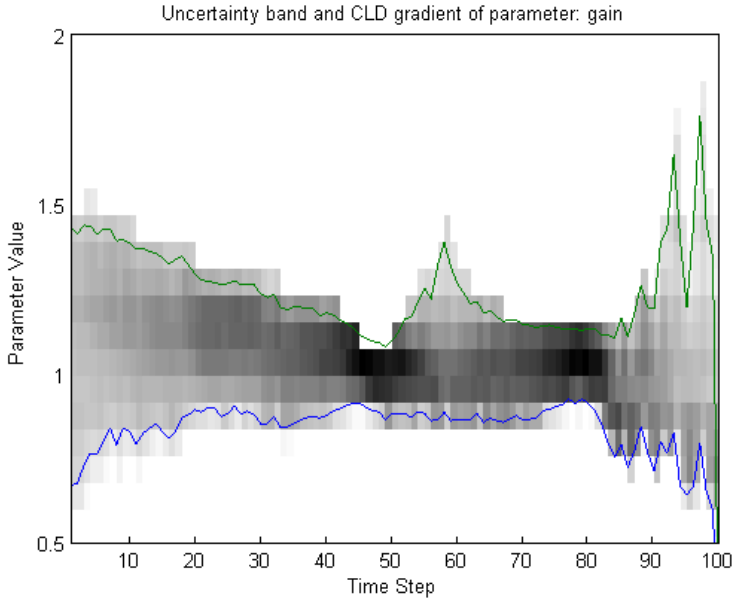

Figure 2.9: Parameter estimation with DYNIA for the same parameter as in Fig. 2.3. Darker color means larger gradient. The true parameter value is 1.

3 Implementing DYNIA

A goal of this thesis is to produce and present an implementation of the DYNIA method. There is an implementation of the method in the MCAT toolbox for Matlab [33], but the source code of the implementation is not available. A presentation of the implementation is included to provide further insight to the functionality of DYNIA, in addition to giving ground for a discussion and research on possible modifications to improve its functionality. Some of the Matlab files that were created for the implementation and the examples are found in Appendix B. All are delivered digitally to the faculty at the time of submission of the thesis.

The flowchart in Figure 3.1 gives a general overview of the steps that are taken in the developed implementation. The first step is to provide all the necessary data that the DYNIA algorithm rely on. This includes obtaining a data set with time, input and output values, the choice of a model structure with parameter ranges for each parameter, and the selection of the number of Monte Carlo samples, the window size, the number of bins used in the CLD gradient calculation, and the percentage x of the best performing parameter sets that are to be used in the moving-window calculations.

The Monte Carlo sampling is performed by simulating the model structure for each sampled parameter set. The whole time-series of the simulated output is stored for each of these parameter sets. In this implementaton the simulations are taken care of by using the Matlab function *lsim* on a system model in the form of a transfer function or a state space model.

When the Monte Carlo sampling and the simulations have finished, the moving-window calculations begin. These are the ones that are iteratively performed for each time step as shown in the flowchart. The Matlab file *movingWindowCalcs.m* presents how this is implemented.

The first step of the moving-window calculation is to calculate the summed square error (SSE) over the window period, for each sampled parameter set. Next, the parameter sets are ranked according to their SSE value and the top x percent performing parameter sets are contained in the behavioural set. x was mentioned as a parameter that is set before the start of the DYNIA algorithm. Alternatively, one could set an upper limit on the acceptable SSE value over a window period. The selection of the best performing parameter sets for a window is implemented in *selectBestParameters.m*.

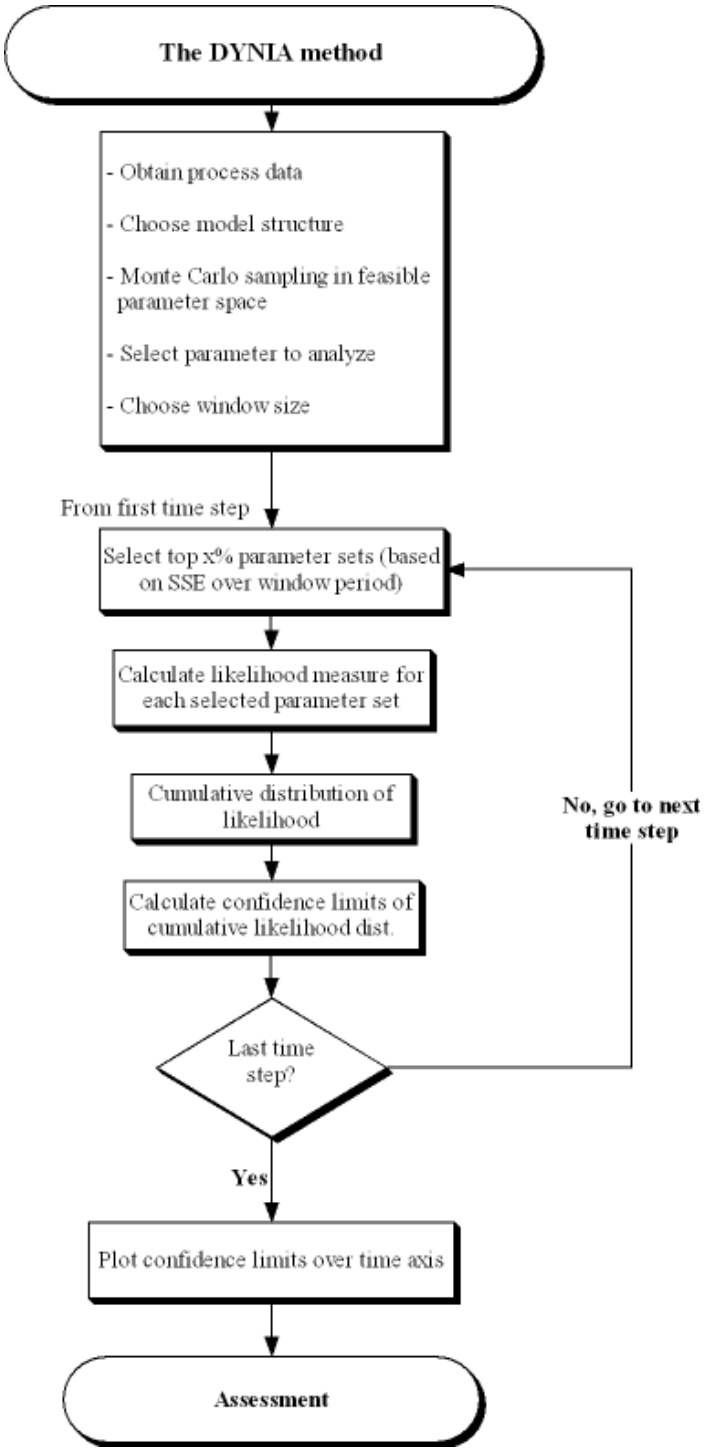

Figure 3.1: A scheme of the DYNIA implementation

Next up is to perform the DYNIA analysis on the behavioural parmeter sets for the current time step (i.e. window). The DYNIA analysis consists of four steps that all are done for each behavioural parameter set for the current time step; calculation of a likelihood measure (*calcLikelihood.m*), calculation of the CLD (*cumulLikelihood.m*), calculation of the gradient of the CLD (*cld_gradient.m*) and calculation of the confidence interval of the CLD (*confLimits.m*).

The calculation of likelihood in this implementation is equal to the method suggested in [33], as presented in Section 2.2.2. The likelihood measure for each behavioural parameter set is then used to calculate the CLD for the parameter being analyzed.

For the range of parameter values that are in the behavioural set, the CLD is calculated by sorting the parameter values and then cumulatively adding the likelihood measure for each sampled parameter value. With the likelihood measure that is applied this give a CLD that is monotonically increasing to a value of one. Parameter samples that have a value below the lowest value in the behavioural set are given a CLD value of zero, while parameter samples with a value above the highest value in the behavioural set are given a CLD value of one.

The third main step of the moving-window calculations is to calculate the gradient of the CLD. The feasible parameter range that is set before running the Monte Carlo sampling is divided into bins of equal size, the number of which was also set in the preamble. If all parameter values in a bin is lower or higher than the lower and upper limit of the paramter values in the behavioural set, the gradient of this bin is zero since the CLD will have a constant value throughout this bin. Otherwise, the gradient value for a bin is calculated as the difference between the CLD value at the beginning and the end of the bin. The number of bins relative to the size of the feasible parameter range is thus obviously the deciding factor with respect to the resolution of the gradient calculations.

Finally in the moving-window calculation, confidence limits are calculated for the CLD. Upper and lower confidence limits are obtained by calculating the upper and lower index that correspond to the interval that encompass a central percentage of the data. This percentage is decided by the choice of an uncertainty level. The parameter values of the confidence limits are then found at these indexes from the sorted vector of the behavioural parameters. After this, the information content for the parameter under analysis, at a given time step, is calculated as shown in Eq. 2.3. Information content, confidence limits and the gradient for each parameter bin is saved for each time step.

When the moving-window calculations have been completed for all time steps, presenting the gathered knowledge to the user is the next task. The code for the generation of these plots are found in the files that are used to run the examples in the next section.

3.1 Examples

Two examples and one case study are presented in this section, all of which have been performed with the presented implementation of DYNIA. The examples are included to give a general insight into the functionality of the implementation, and to assess its performance to a known model. The case study uses real process data from an offshore oil production facility that uses the PIMAQ system of Siemens for management and acquisition of process data. The DYNIA method is then applied to this data set.

3.1.1 Example 1: First Order System

The first example consider a first order linear time-invariant (LTI) system. The time constant and gain parameters characterize the exponential response to step changes on the input. Although first order systems describes dynamics that are simple of complexity, many control systems in the oil and gas production industry can be described well with a first order model.

Input data were chosen to be constant for large parts of the data set, with some step changes inbetween. From the experience of the author this is also the case with control systems in many real-life scenarios. The output data was generated by:

$$\frac{y}{u}(s) = \frac{k}{T_1 + 1} \qquad k = 2,\ T_1 = 3 \tag{3.1}$$

Some noise was added to the output signal to imitate measurement noise. The input and output data for the experiment is show in Figure 3.2.

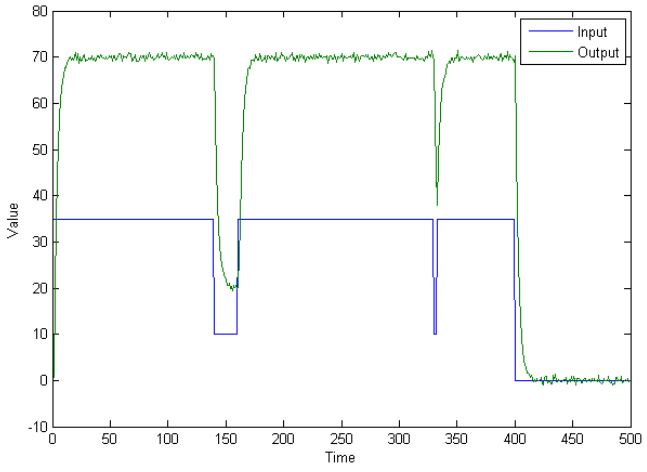

Figure 3.2: Input/output data for this example. A first order system

This data set was then subject to an analysis with the DYNIA method. First, the preliminary conditions had to be decided. This includes choice of model structure, setting parameter ranges and deciding window size number of Monte Carlo samples. The model structure was selected to be corrspondent with the "true" one, a first order LTI system. The parameter range for both parameters was set to be $\theta \in [0, 7]$. The example can be run in Matlab with the file $example\_firstorder.m$.

As can be seen from Figures 3.3 and 3.4, the gain parameter has a high information content and is easily identifiable whenever there is a constant input value not equal to zero. This is as expected, since the gain describes the relationship between the input and output when the system is in a stationary, non-zero condition. As we can see from the plots, the data set has a high information content with regards to the gain parameter until the system goes to its zero condition.

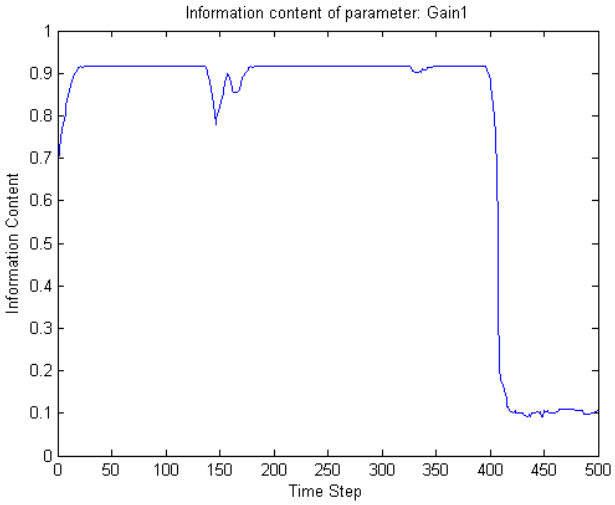

Figure 3.3: Information content of the gain parameter

The next two plots, Fig. 3.5 and 3.6 show the corresponding information for the time constant. It is observed that the information content of this parameter is very close to being the exact opposite of that belonging to the gain parameter. This is fairly obvious since there can not be made a decision about the time constant as long as the system is in a stationary state. The last input step puts the system in a stationary zero state. Consequentially, from that point and out, the data set contains information regarding, in its near entirety, the time constant. It is observed how the information content steadily increase and the parameter uncertainty steadily diminish after the input step to zero. This goes in line with the thought that the more (informative) data, the better.

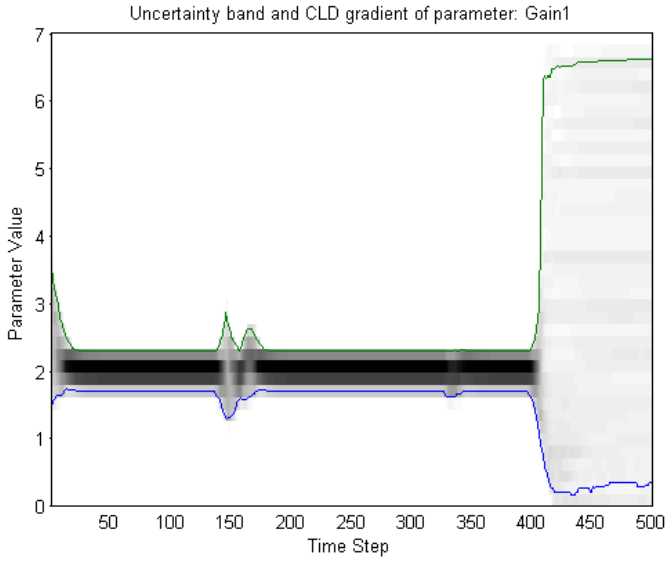

Figure 3.4: Parameter identifiability plot of the gain parameter

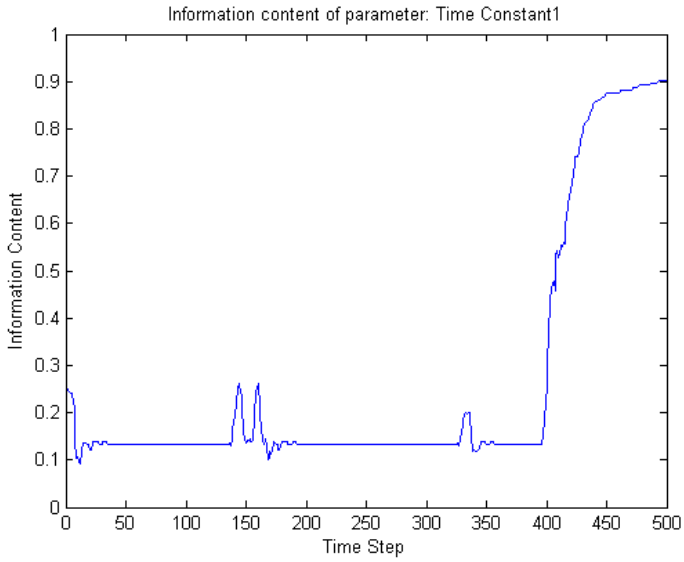

Figure 3.5: Information content of the time constant parameter

In addition to this, there are small spikes of higher information content for the time constant around the two short input step changes. But, with the noise on the output signal, the short duration of the input steps before

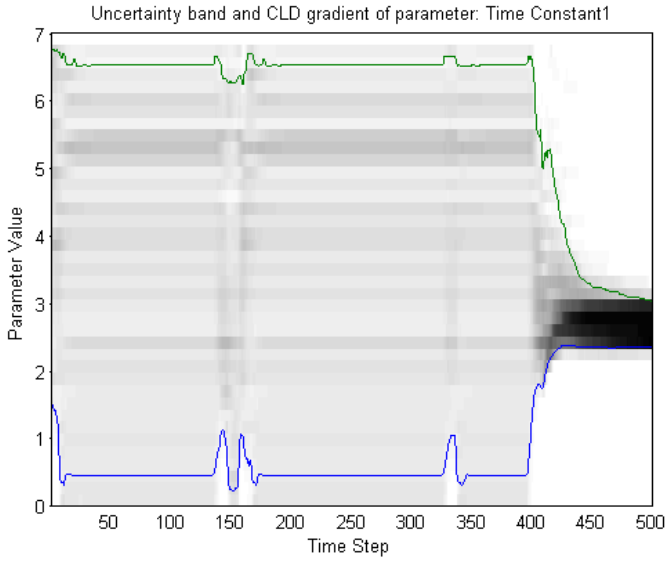

Figure 3.6: Parameter identifiability plot of the time constant parameter

they return to their previous value and the relatively small number of Monte Carlo samples (2000) compared to the large parameter ranges, the parameter uncertainty is still large. However, these spikes in the information content are not "worthless". They do indicate segments of the data that are interesting with regards to this parameter. Knowing this, there are numerous options on how to proceed. One option is to select a data segment around a spike and use it directly as the data set in a traditional system identification method. That would surely work well in this case, but the low peak, i.e. high parameter uncertainty, of the spikes imply a risk that the segment may not describe this parameter well after all. Alternatively, one could use the DYNIA method again to reduce this risk. One option is to simply icnrease the number of Monte Carlo samples. This will always improve the performance of any sampling-based uncertainty analysis, one of which the DYNIA method is based upon. The computational cost is likely to increase by a large amount though, rendering this option less attractive. A more enticing approach is to reduce the parameter ranges for the Monte Carlo sampling for the parameters that are well identified. The first run of the DYNIA method gave a very strong and unique indication on the value of the gain parameter. By minimizing the feasible range for the gain parameter, a new run of the DYNIA method on the data set give increased performance with regard to the other parameters, in this case the time constant. Figures 3.7 and 3.8 show the DYNIA output for the time constant when a much tighter range on the gain parameter is applied.

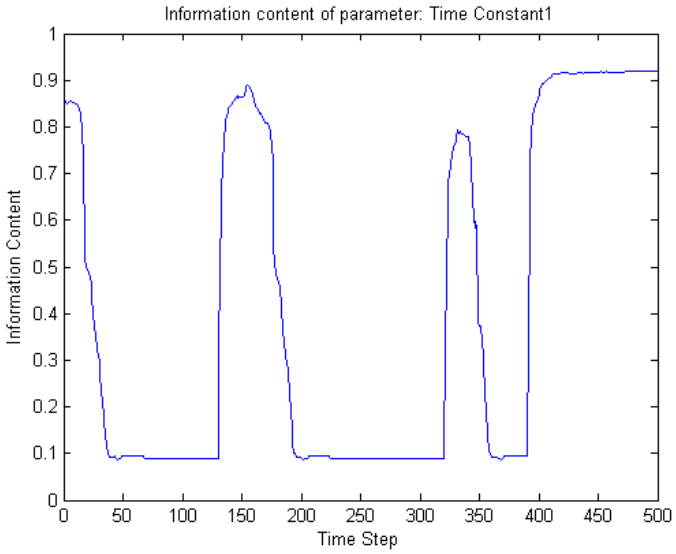

Figure 3.7: Information content of the time constant parameter after the second run

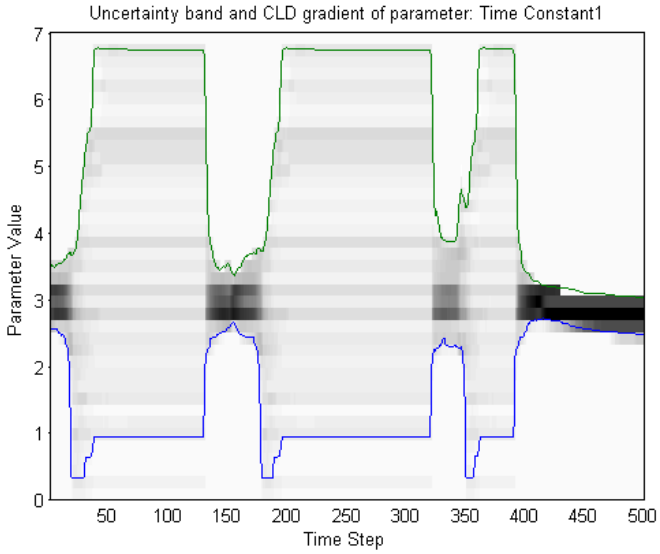

Figure 3.8: Parameter identifiability plot of the time constant parameter after the second run

3.1.2 Example 2: Second Order System

In this example the system output was generated by the following overdamped second order LTI model with time delay:

$$\frac{y}{u}(s) = \frac{k}{(T_1 s + 1)(T_2 s + 1)} e^{-\Theta s} \qquad k = 2, T_1 = 2, T_2 = \frac{2}{3}, \Theta = 100 \tag{3.2}$$

All the parameters, i.e. the gain k, the time constants T_1 and T_2 and the time delay Θ, were subject to the DYNIA analyis. Figure 3.9 show the input and output data. As can be seen, the data set is in its zero state most of the time, with two input steps of short duration. It is also obvious that there is a significant time delay and considerable noise on the output signal. In this example, as in the previous, the model structure that the DYNIA method was performed with was the same as the one used when generating the output. The parameter ranges, which correspond to the y-axis intervals of the parameter identifiability plots, must also be considered as quite large.

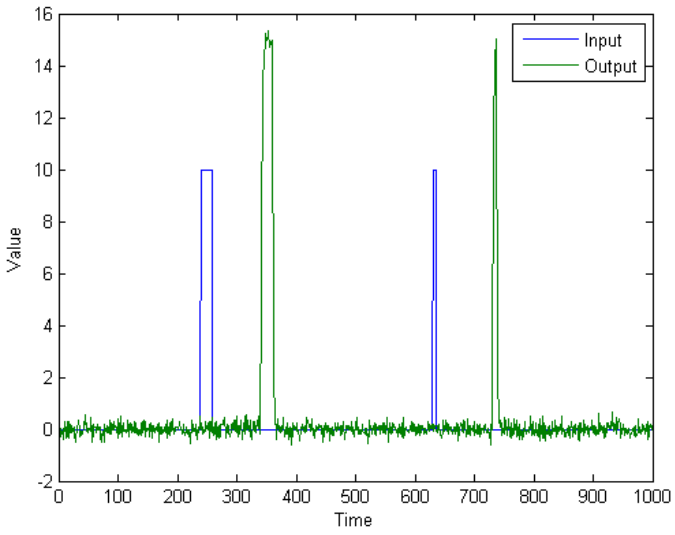

Figure 3.9: Input/output data for this example. A second order system with time-delay

Beginning with the time delay parameter, we see that this parameter has a high information content and is well identified at the time when an input change acts on the system output (Figures 3.10 and 3.11). It is important to notice here that the temporal variance of information content and parameter uncertainty are calculated and plotted with regard to the time when this information is discovered. In the case of the time delay this means that a

strong estimate on the time delay was not first found when the first input change was applied, but when the first input change actuated the system output.

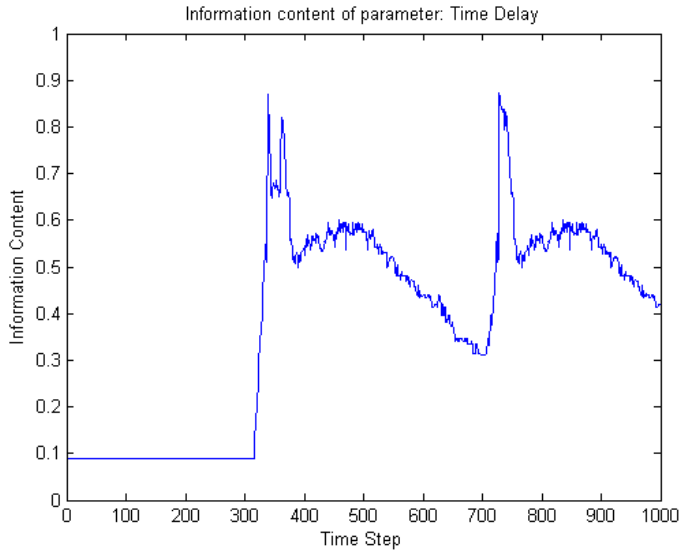

Figure 3.10: Information content with regards to the estimation of time delay

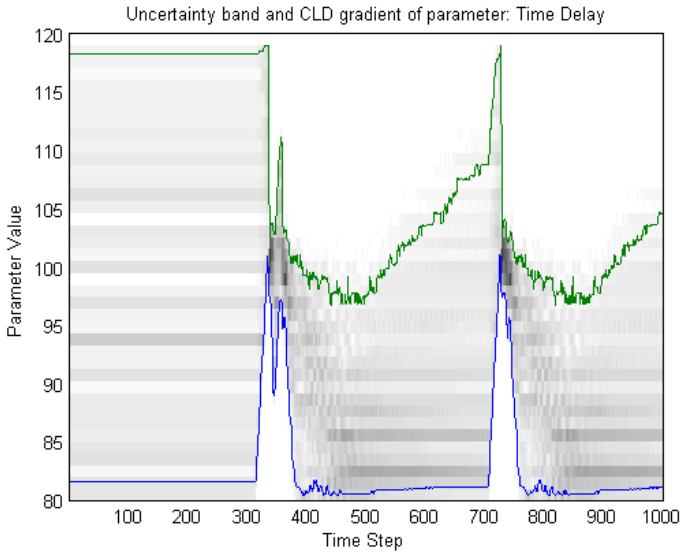

Figure 3.11: Parameter identifiability plot of the time delay parameter

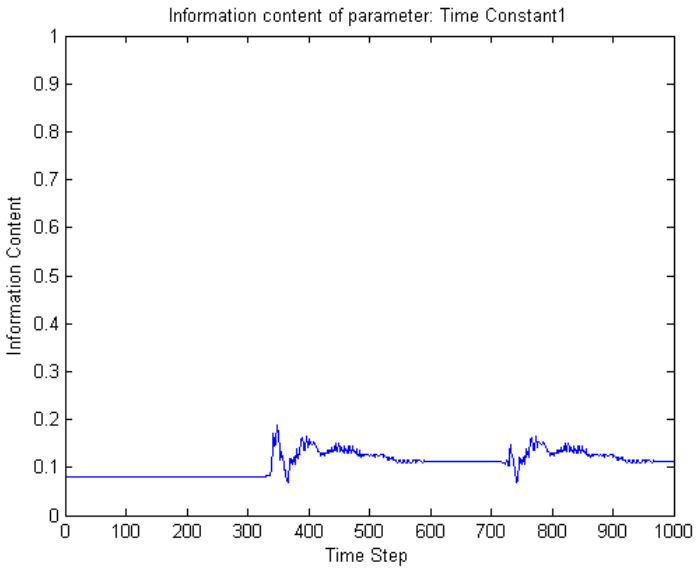

Figure 3.12: Information content of the first time constant parameter

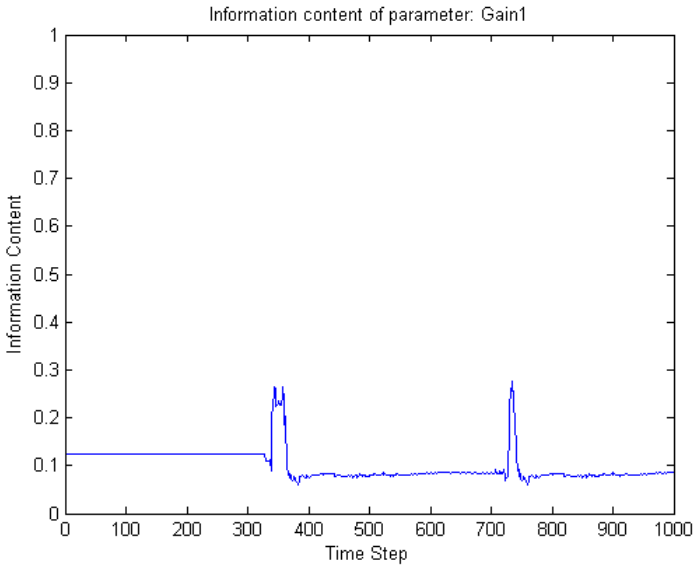

Figure 3.13: Information content of the gain parameter

When selecting data segments in systems with time delay with the DYNIA method, it is important to keep in mind that the input data corresponding to high information content for a certain parameter might occure a considerable amount of time (i.e. one time delay) before an increase in information content is discovered.

This second order system has two time constants. Figure 3.12 show the DYNIA analysis of T_1. At this stage in the process, with the given choices of parameter ranges etc., this parameter is poorly identified by the DYNIA method, and the plot of the information content does not give reliable results for data segment selection. As will be seen later in the example this does not mean that it is not possible to get a stronger indication on data segments that carry information with regards to this parameter.

The second time constant, T_2, show substantially better properties with respect to information content and parameter identifiability than T_1(Fig. 3.14 and 3.15). The parameter identifiability plot show that the gradient of the cumulative likelihood distribution is considerably larger for the lower part of the uncertainty interval. Looking at the model that generated the output this corresponds very well with a true parameter value of $\frac{2}{3}$.

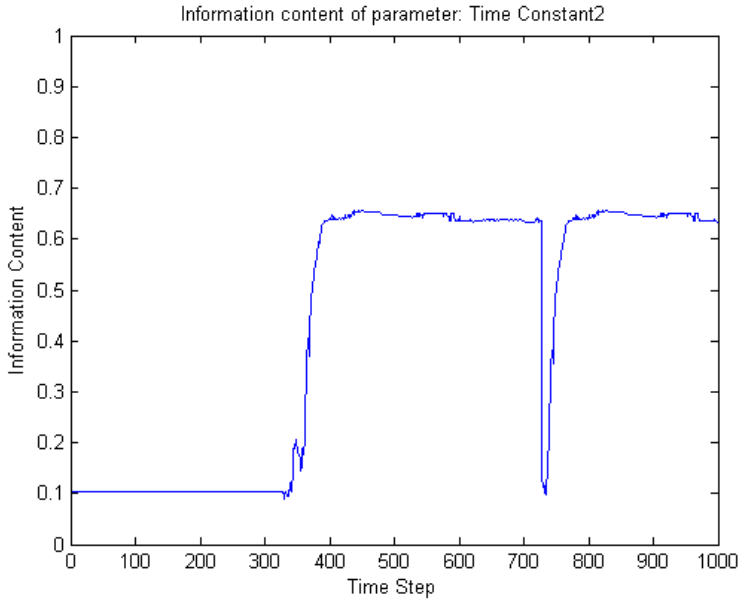

Figure 3.14: Information content of the second time constant parameter

The last parameter to be analyzed is the gain, k. From Figure 3.13 we see that there are tendencies to increased information content when the input actuates the output, but in a real-life scenario these results surely would have been likely to be considered inconclusive. So, after performing the DY-

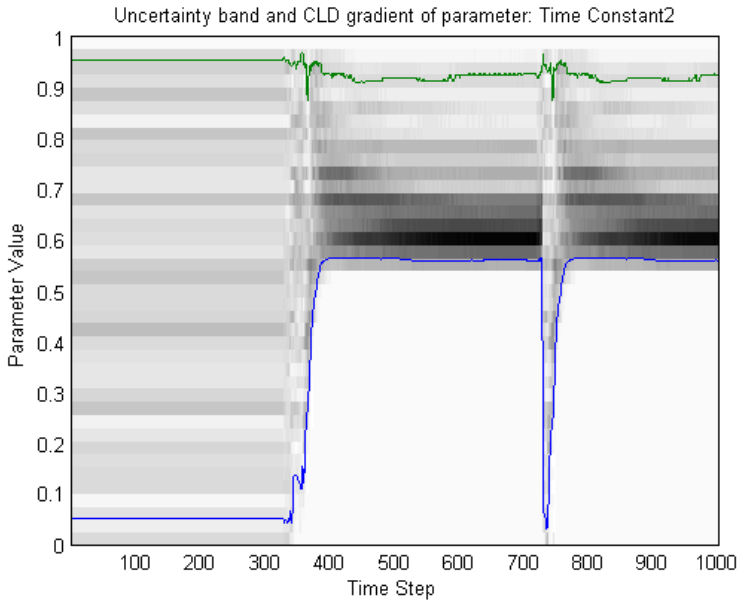

Figure 3.15: Parameter identifiability plot of the second time constant

NIA analysis on the four parameters, with fairly large parameter ranges one should add, we have two parameters, Θ and T_2, for which a good amount of information in the data set has been discovered, and two, T_1 and k, for which this not has happened. As mentioned in the previous example, the DYNIA method is highly dependent on the number of Monte Carlo samples. However, narrowing down the parameter ranges for the parameters that have been found to have a strong indication on being well identified will also improve the performance of the method for the parameters that have not yet been properly identified.

Figures 3.16 to 3.19 show that this indeed is the case in this example when stricter parameter ranges are set for the parameters that were well identified after the first run of DYNIA.

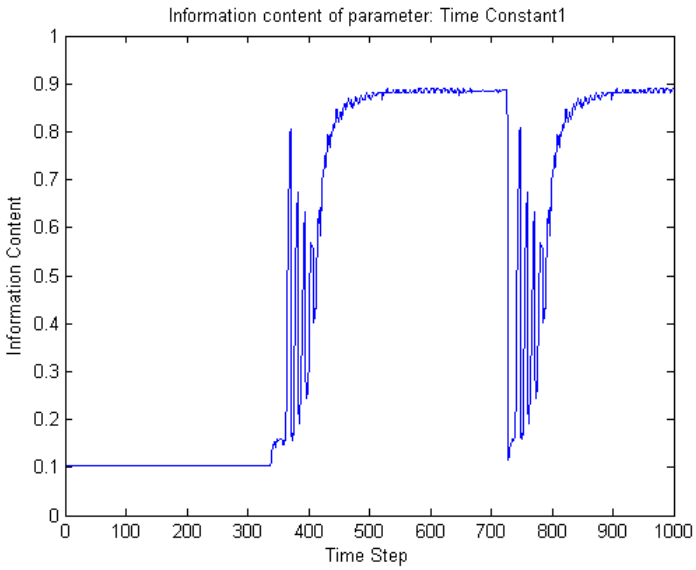

Figure 3.16: Information content of the first time constant on the second run of DYNIA

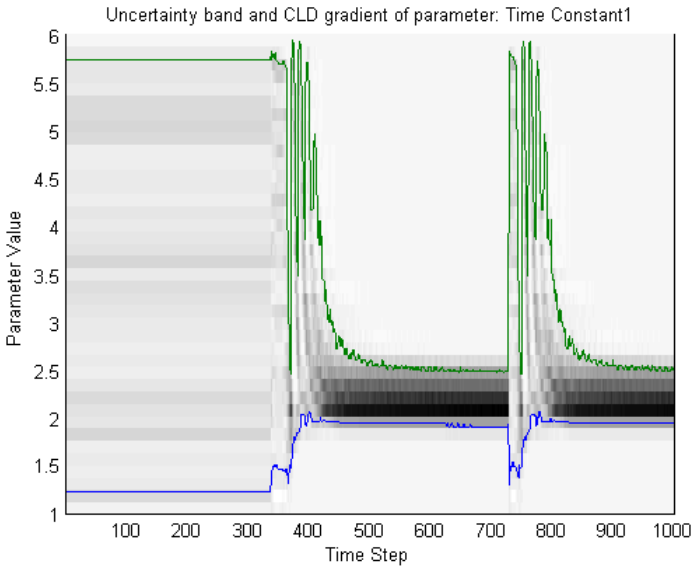

Figure 3.17: Parameter identifiability plot of the first time constant on the second run

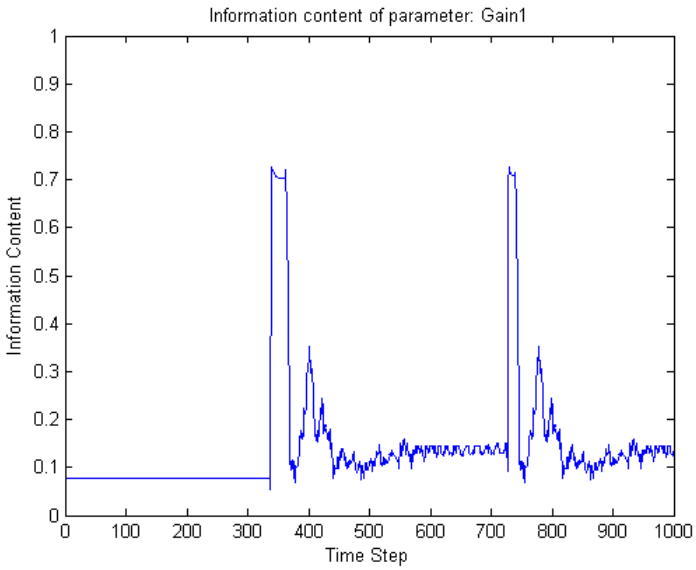

Figure 3.18: Information content of the gain on the second run

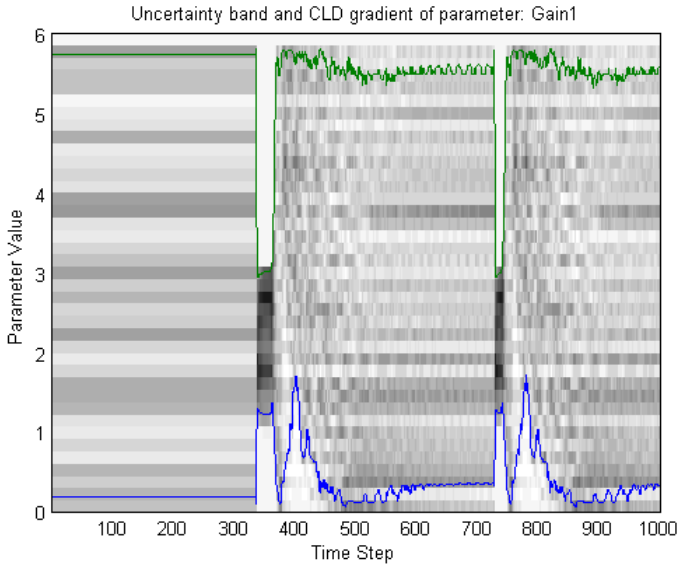

Figure 3.19: Parameter identifiability plot of the gain on the second run

3.1.3 Case study

A case study has been performed in order to test the impementation of DYNIA with real process data obtained by the PIMAQ system of Siemens. PIMAQ is an advanced process information management and acquisition system that has storage capacity for the process data generated throughout the lifetimes of oil and gas production systems. The data in this case study stems from the Al Shaheen oil field, which is a part of the Dukhan field which is the largest producing oil field on the west coast of the Qatar peninsula. Al Shaheen is operated by Maersk Oil Qatar AS, producing 200000 barrels per day.

The data set used in this example consists of input and output data obtained from the control of CWR (cooling water return) in a cooling water system. Details of the specific system herein considered is not given, the data is purely used to analyze its information content. Generally though, the objective in treating heat exchangers that use cooling water as a heat discharge fluid is to keep the water side as clean and corrosion free as possible. Open recirculating cooling systems are prime candidates for contamination problems. If left untreated, contaminants are allowed to concentrate in the system as the cooling water evaporates. This can lead to numerous problems, e.g. scale, fouling, microbiological growth and corrosion. These problems can among other things cause breakdown of the metal parts of the cooling system and reduced heat transfer efficiency. Loss of heat transfer efficiency can cause reduced production. In fact, if the heat transfer falls below a critival level, the entire system may need to be shut down and cleaned. For these reasons the temperature and pressure of the cooling water that is returned into the system is controlled.

The input and output values are shown in Figure 3.20. The data set consists of approximately one months worth of data. A challenge for the use of the DYNIA implementation on this data set was that the sampling done by PIMAQ is event-based. Thus, unlike in the previous examples, there is not a constant time step between samples. In addition, the input and output values were sampled independently, so their time vectors were not equal. It was found necessary to preprocess the data to get a common time vector for the input and output data. This was done by the use of linear interpolation. It was observed that the sampling interval for both the input and output is very small when the system is actuated in some sort, i.e. when something is happening. Consequently, the interpolation and "synchronization" of data occured predominantly in parts of the data set with steady input and output values.

The event-based sampling led to some problems with respect to the simulations of the system, and the DYNIA implementation that has been presented in this thesis is more applicable to systems with a constant sampling interval. Nevertheless, the implementation was used on the data set and promising results were obtained.

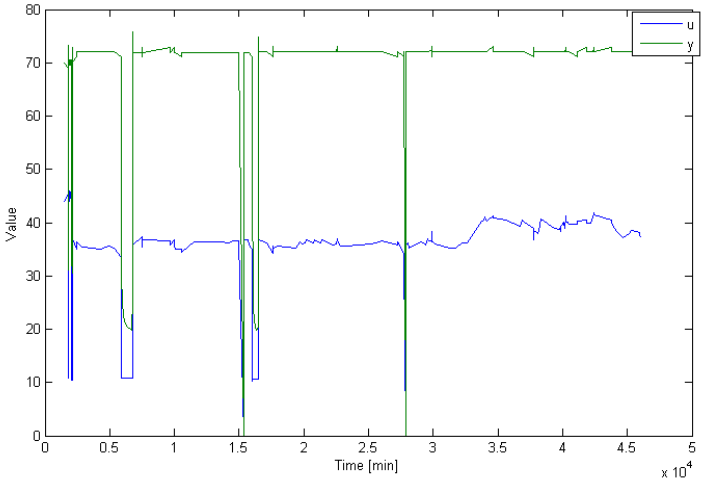

Figure 3.20: Input/output values of the data set used in the case study

The first step of the case study was to create a baseline model of the system in order to be better positioned to analyze the results from the DYNIA method. This was done in the traditional manner of manually searching for input steps and using the data around these steps to calculate a model based on the step response. A first order model of the system was obtained:

$$\frac{y}{u}(s) = \frac{k}{T_1 + 1} \qquad k = 2.04,\ T_1 = 2.16 \tag{3.3}$$

This model was calculated by fitting the data around the input change at time $1.6x10^4$ to the step response of a first order model with the method of least sqaures.

The parameter ranges were set to $k \in [0, 6]$ and $T_1 \in [0, 8]$. It was observed that the system had quick responses to input, and it was found appopriate to use very small window sizes, with n values of 1-3 (window size $= 2n+1$). The percentage of the sampled parameter sets giving the lowest summed squared errors for each window was set to 10%. This has been suggesteg as a decent choice in the existing literature on DYNIA [33]. This example was performed with 5000 Monte Carlo samples. Generally, more is more when it comes to Monte Carlo sampling. This is also the case for DYNIA, but there is an obvious cost in increased computing time.

Figures 3.21 and 3.22 show that k has a strong indication on being well identified for most of the time series. The information content is generally high all over, with a few negative spikes that coincide with step changes on the input. This is not surprising in this case where the input is close to constant for most of the data set, given that the gain parameter will determine the size

of the steady state response when the input settles to to a constant value.

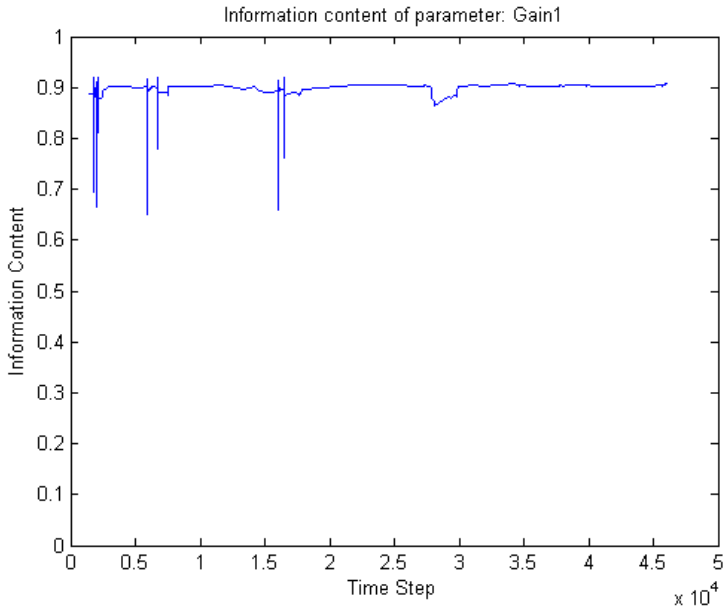

Figure 3.21: Information content of the gain parameter

The parameter identifiability plot for the gain parameter (Fig. 3.22) show how the parameter uncertainty and the gradient of the cumulative likelihood distribution varies with time. This plot give a strong indication on that this parameter is well identified with a parameter value of $k \approx 2$. Notice however that the gradient of the CLD has its highest value (i.e. is darkest on the plot) for a value of $1.7 - 1.9$. Taking a look at the input/output plot in Figure 3.20 we see that the input actually has increased quite a bit from a value hovering around 36 to around 40. This happens without any noticeable change of the output value, which remain fairly constant around a value of 72. This could indicate that the system has gone through a change of operational mode, or that something else has impacted the physical attributes of the system dynamics. This is an example of how the DYNIA method can be able to pinpoint the time and magnitude of changes that for some reason, known or unknown, has been made to the system dynamics.

Figure 3.23 show the information content for the time constant T_1, unfortunately with very little indication towards data segments with healthy amounts of information with respect to the parameter. There are some minor spikes that, when compared to the I/O plot, do correspond to the times when changes to the input are made, which is also when you would expect informative data for a time constant. The results are however not of a quality that makes one able to draw any kind of conclusions towards whether or not this

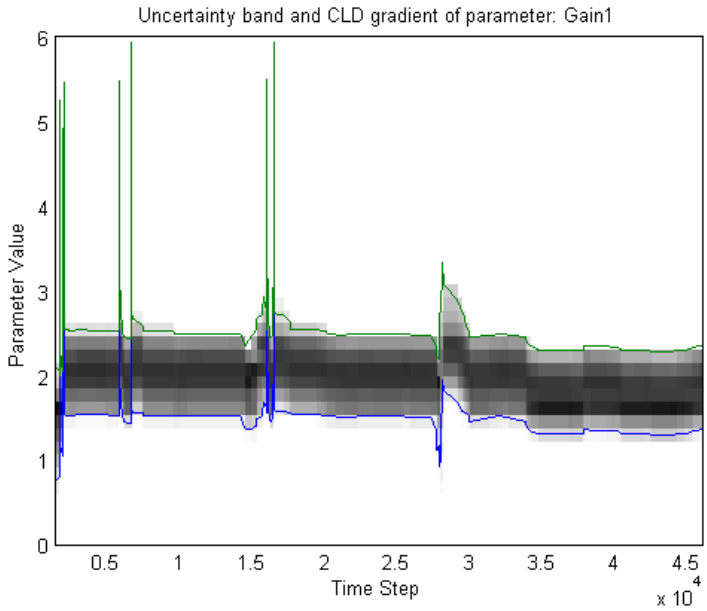

Figure 3.22: Parameter identifiability plot of the gain parameter

data set is informative with respect to the time constant.

What we have seen in the two previous examples though, is that the quality of the results from a DYNIA analysis for a parameter can improve dramatically when knowledge about the other model parameters is used to improve the performance of the uncertainty analysis. The gain parameter k was shown to be strongly identified for this system. After setting a strict parameter range of $k \in [1.8, 2.2]$, a new DYNIA analysis is performed for the time constant.

The results from the second run of DYNIA for the time constant T_1 are shown in Figures 3.24 and 3.25. The results have improved strongly. The segments of high information content are considerably more pronounced than after the first DYNIA analysis. This clearly shows the value of exploiting existing knowledge of the system with regards to improving the performance of DYNIA.

Looking at the parameter identifiability plot (Fig. 3.25), we see that the time with the highest information content corresponds with a parameter value close to 0. Looking at the I/O data again, it is clear that at this time there is an erroneous reading in the data set where both the input and output values are zero for one sample before they return to their previous steady state value. It is unfortunate that this erroneous gives the highest measure on information content, and this is something the user has to keep in mind when using the DYNIA method. Consequently, it is sensible to check that the data segments with high information content does not contain erroneous data.

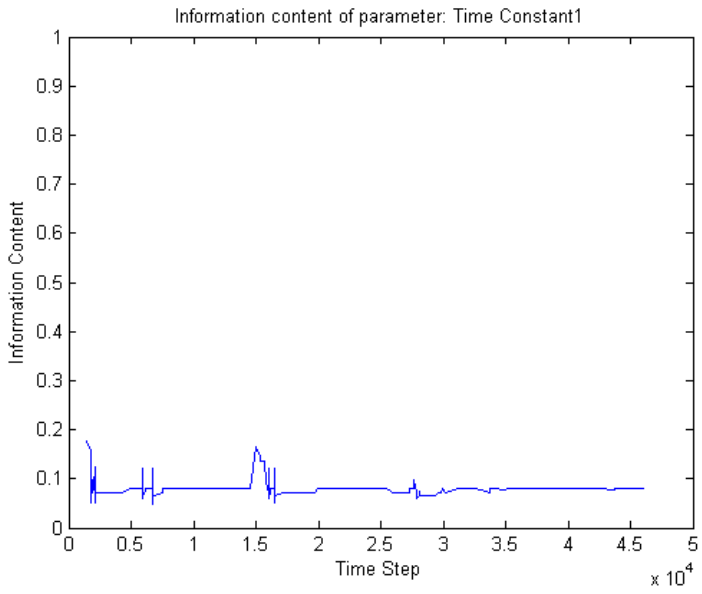

Figure 3.23: Information content of the time constant parameter

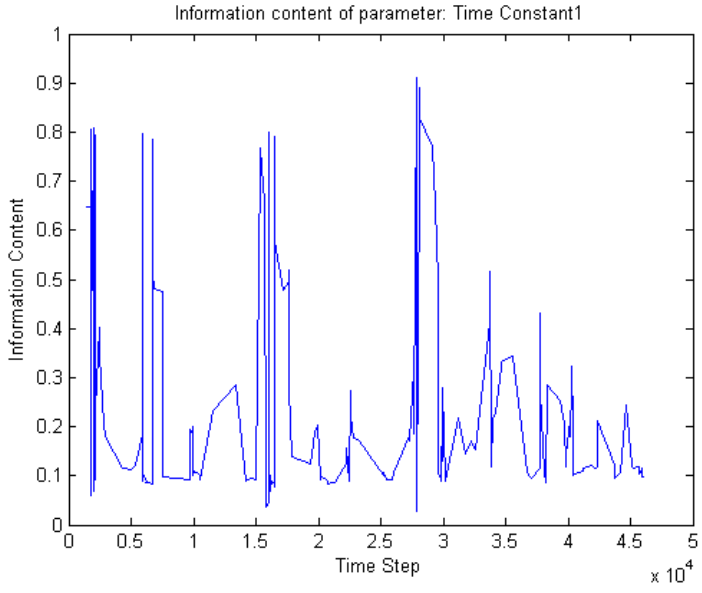

Figure 3.24: Information content of the time constant after the second run of DYNIA

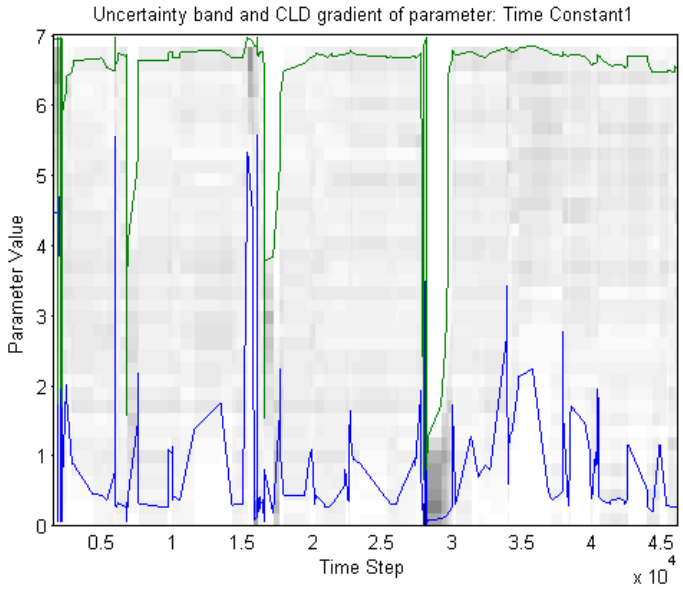

Figure 3.25: Parameter identifiability plot of the time constant after the second run

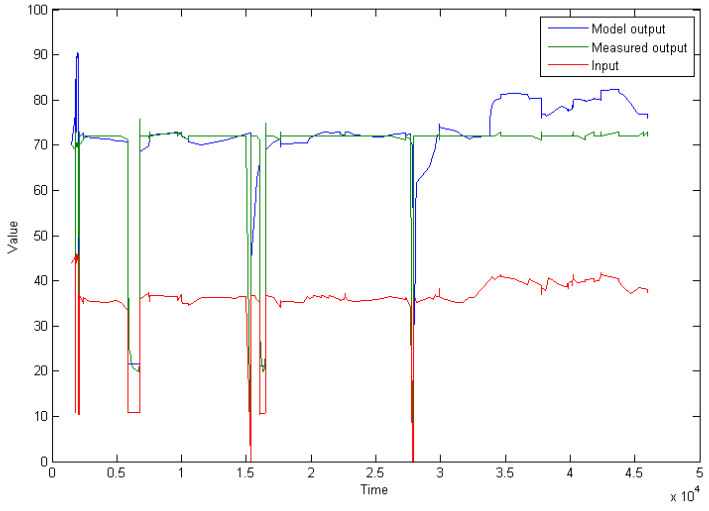

Figure 3.26: Comparison between modelled and actual output. $T_1 = 2$, $k = 2$

The other major spikes on the plot of information content correspond to input step changes, thus being reasonable with regard to where one would expect to find informative data for the time constant in this data set. Looking at the parameter identifiability plot, the quality of the results is not good enough to give a strong indication on the parameter value. However, the most probable reason for the use of DYNIA is to indicate data segments that may be prosperous when used as the data set in a system identification procedure. In that context the DYNIA analysis has provided promising results for this case. Even so, the parameter identifiability plot does give indications on a time constant value in the range of $1 - 3.5$. Different values from this range were used to simulate the system and compare it to the real value. Figure 3.26 show one such comparison between simulated and actual output. It became clear that the different time constants gave very similar overall model performance as measured by SSE, the gain was kept at $k = 2$ for all simulations:

Time Constant	SSE
1	$1.8803x10^4$
1.5	$1.8776x10^4$
2	$1.8772x10^4$
3	$1.8772x10^4$
4	$1.8772x10^4$

4 Industrial Applicability of DYNIA

This chapter considers how different aspects related to the DYNIA method impacts the practical applicability in the industry.

4.1 Window Size Selection

We have seen that parameter identifiability is calculated at each time step using the residuals for a number of time steps, n, before and after the point considered. The number of time steps that should be considered in each window is dependant not only on the response time of the parameter under investigation, i.e. on the period over which the parameter is influential, but also on the quality of the data set [32]. An unappropriately large window size is likely to mix periods of noise and information, blurring the information. Running the examples from the previous chapter with increasingly large window sizes also gave clear indications on this. A suggestion on the choice of n is to use the parameter range selected for the dominating response time parameter as a starting point.

In general, smaller window sizes give better results with respect to the conservation of information, but also increase the risk of obtaining results that can be distorted by measurement errors and the like. This obviously depends on the quality of the data. Knowledge of measurement uncertainty and other aspects on the quality of the data set should be taken into consideration when the window size is selected.

A note should also be made on the results that are obtained for some of the first and last points of time in a data set. With DYNIA, the moving-window approach implies that for the n first and last data points, a full window-size is not possible. Hence, the results from these parts of the data set will to some degree be distorted, and care should be taken when they are analyzed. Decisions made from the DYNIA method should at least not solely be based upon these results.

4.2 Computational Complexity

The examples in the previous chapter show that the DYNIA method in fact has the capability to analyze how information content and identifiability, with respect to a specific parameter and a model structure, vary with time. On the other hand, they also demonstrate that good results might not be achieveable without some effort from the user. It appears that to achieve a good DYNIA analysis for all the parameters of a system, it will often be necessary to analyze one parameter at a time and apply conclusions drawn from one DYNIA instance for a parameter into a new instance for another parameter. The size of the parameter ranges for the Monte Carlo sampling is especially decisive in this matter, and it is connected with the decision of the number of Monte Carlo samples to perform and the number of parameter sets to contain in the behavioural set. If the DYNIA method is performed with wide parameter ranges and a relatively low number of samples, the method will on many occasions be doomed to give at the best weak indications towards segments of high information content for a parameter. The problem when this is the case is that the sampling density is low, and that it impacts and impairs the ability of the method to "see" how the information content vary with time. This can be compared with the impaired ability to see details of an image when the resolution is reduced.

The sampling density, d, for one specific parameter is decided by the relationship between the number of samples of that parameter, s, and the size of the parameter range, p. The number of samples to achieve a desired density is thus the product of the density and the parameter range.

$$\begin{aligned} d &= \frac{s}{p} \\ s &= d \times p \end{aligned} \tag{4.1}$$

If we assume that the parameters of a system have similar demands for the number of samples required to achieve their desired densities, we find that the total number of samples, S, required to maintain these densities when the parameters are put together in parameter sets grows exponentially with the number of parameters, n:

$$S = s^n \tag{4.2}$$

Each sampled parameter set is simulated at all points in time, nT, in the data set, for a total number of $nT \times S$ evaluations of the algorithm used to solve the differential equation of the system. Since the exponential growth is the dominant factor as n grows larger, the parameter sampling and simulation has exponential time complexity when considered by the big O notation, when the sampling density is to be constant and irrelevant of the number of parameters:

$$T(n) = O(2^n) \tag{4.3}$$

The parameter ranges can in many cases be reduced, the paramteters may not need equally many samples to achieve desired sampling density, and processing power may be increased, but none of these matters can cover up for the fact that the problem is of exponential complexity. For the practical use of DYNIA on systems with more than a handful of parameters, this is of huge impairment. However, the examples have shown some promising practical functionality, and the following section will display a suggestion on how the DYNIA method could be included in a process data logging system to achieve better practicability of the method.

The moving-window calculations of the implemented version of DYNIA evaluates the SSE of the output for the simulations for each sampled parameter set, compared to the real output, to obtain a behavioural set of parameters. This has linear time complexity. Likelihood and CLD of the behavioural parameters are then evaluated with linear time complexity. Since the size of the behavioural set is dependent on the total number of samples, the big O notation for the time complexity of the moving-window calculations, with S samples, becomes:

$$T(S) = O(S^2) \tag{4.4}$$

4.3 Practical Inclusion of DYNIA in a Logging System

The suggestion is to run an instance of the DYNIA method that analyze the process data while these data are obtained by the logging system. The main thought is that the Monte Carlo sampling is performed once, and that the moving-window calculations are done in a "semi-online" way as the logging collects new process data.

The idea is that when new input and output data is logged at a point in time, one iteration of the moving-window calculations described in Chapter 3 is performed. Depending on the window size number n, the moving-window calculations are performed for the data logged at $t - (n + 1)$, where t is the current time step. This includes simulation of the Monte Carlo sampled parameter sets to obtain the behavioural set that is to be used for the window under evaluation. The system will only have to be simulated for the time steps that are a part of the window, something which should be possible to do effectively. Then, likelihood values, cumulative likelihood distribution (CLD), gradients of the CLD and uncertainty intervals are calculated, and the values for information content and parameter identifiability are stored. In this way it is possible to see how the information content in the data vary as the data is being logged.

When the DYNIA method is applied in this way, a database with the time-varying information content is readily available when a decision is made to reevaluate the model parameters by the use of the system identification procedure. In other words, this way of including DYNIA in a logging system nearly eliminates the overhead cost and labour that is introduced by stopping the production to perform specificially designed experiments. It is also more time-efficient than performing the DYNIA method in the traditional manner, as per the examples from the previous chapter.

Based on how the information content of the different system parameters vary with time, the obvious approach in the search for an informative data set is to base the system identification procedure on data segments where high iformation content has been suggested by the DYNIA method.

The data segment selection could be automatic, by introducing a lower limit on the acceptable value of the information content measure. Based on the experiences drawn from the examples though, it may seem like a subjective selection based on plots of information content plots is the most appropriate approach, at least with the current implementation of the method.

4.4 Including DYNIA in the System Identification Procedure

Based on the properties and possibilites with DYNIA it seems like the method can be used in a system identification procedure, either solely as a tool for localizing data segments of high information content, or as a validation method.

A suggestion on a system identification procedure could be to first employ the DYNIA method on the process data set. Then, based on some criteria, data segments of high information content are chosen as the data set to be used in the assessment of the candidate models (the parameter estimation in this case, since a globally identifiable model structure is assumed). This assessment could take place with a parameter estimation technique such as the least-squares method, described in Section 1.2.3.1. The parameter estimates generated by the DYNIA method could then be used for some sort of verification of the results.

A prerequisite of both DYNIA and traditional system identification techniques is the selection of a model structure that the analysis is carried out on. A priori knowledge of the system is usully the basis on which the model structure is selected. An asset of the DYNIA method in this context is that it has the capability to indicate model structural failures. If the time-variation of the gradients of the CLD, i.e. the measure of the parameter identifiability, varies with time, this should be investigated as a possible sign of the selected model structure not being able to properly represent the system dynamics. This assumes however that the specific parameter does not describe time-varying characteristics of the system response.

5 Modifications to DYNIA

An advantage of the DYNIA method is that it can be used with any appropriate objective function, likelihood construction method, uncertainty interval etc. This means that the method easily can be modified if better approaches in one of these or other areas connected to the method are found. This chapter introduces concepts that are thought to have good opportunites with respect to an increased and more efficient functionality of DYNIA.

5.1 Parallel Computing

Parallel computing is a cost-effective method for the fast solution of copmutationally large and data-intensive problems. The principle behind parallel computing is that large problems can often be divided into many small problems that can be solved in parallel. Inexpensive parallel computers such as desktop multiprocessors and clusters of PCs have made methods for parallel computing generally applicable [17]. Parallel computing will now be introduced as a benefitial modification with respect to the efficiency of the moving window calculations, and also to the Monte Carlo parameter sampling.

Even though an evaluation of the simulation results for all the time steps that make up the current window is performed in order to obtain the behaviour parameters at that specific time step, this does not mean that the moving window calculations of one time step depends on these calculations done at another time step, neither before or after the time step being under analysis. The simulations for all the sampled parameter sets are performed before the moving window calculations begin, giving the differences between real and modelled output for all parameter sets at all time steps. Thus, the moving window calculations are computationally independent of each other, and they are appropriate for parallel computing.

Parallel computation of moving window should provide large reductions of the total computation time when performing a DYNIA analysis on a system in an "offline" manner, as per the examples. The magnitude of the reductions will naturally depend upon the number of time steps to be analyzed as well as the available supply of parallel computing equipment.

No large modifications to the presented implementation are necessary to

facilitate for parallel computation of the moving windows. However, the initialization of the parallel processes will obviously have to be performed before the commencement of the moving window calculations, and the moving window results from each of these processes must be gathered in an appropriate manner.

Parallel computing can also be applied to Monte Carlo parameter sampling in an intuitive and appealing way, that improves the efficiency of the sampling scheme. We have seen how the DYNIA method need a large number of Monte Carlo samples and simulations to give robust results. With parallel Monte Carlo sampling, we can achieve linear speed-up of the sampling procedure [25]. Suppose it was found that n parameter sets had to be sampled to achieve the desired sampling density. If C computers are available for the sake of parallel computing, each computer need only to produce $\frac{n}{C}$ samples (and corresponding simulations) to achieve the same sampling density as the traditional sampling method. If the computers used for both the traditional and parallel Monte Carlo sampling are assumed to have equal processing power, the parallel MC sampling will have achieved linear speed-up compared to the traditional method.

As with parallel computing of the moving windows, this modification requires small modifications to the implemented version of DYNIA, and the perceived efficiency for the operator should improve well in practice.

5.2 Improving the Parameter Sampling

The DYNIA method has been shown to be flexible and the moving window calculations suitable for parallel implementation. However, the Monte Carlo sampling of the parameter space is not efficient, and a major disadvantage of Monte Carlo sampling is the large number of samples necessary to achieve good density of the samples and thereby reliable answers from the overall method. Parallel Monte Carlo sampling was shown to give linear speed-up, but with the exponential time complexity of the sampling procedure it is important to do more research on the possibility of improving the efficiency of the parameter sampling. Some thoughts on this is provided in this section. Unfortunately, it was found difficult to provide definite suggestions on how this could be done for the DYNIA method, but some guidelines towards what one might focus on in future work is given.

Parts of the DYNIA method builds on aspects from the Generalized Likelihood Uncertainty Estimation (GLUE) technique, which has been described earlier in this text. The Monte Carlo parameter sampling over the feasible parameter space is one of the aspects that has been derived from the GLUE method.

Some research has been put into the investigation of the possibilities to alter the parameter sampling towards better efficiency. [5] discusses the use

of adaptive Markov chain Monte Carlo sampling as an alternative to the traditional Monte Carlo algorithm, while [15] discusses the use of a sampling scheme based on genetic algorithms. These new alternative sampling schemes have shown improved efficiency compared to the traditional Monte Carlo algorithm.

There has also been performed research on the integration of a genetic algorithm in a Markov chain Monte Carlo sampling algorithm, though not with regards to neither DYNIA, GLUE or any other sampling based uncertainty analysis.

It should be said from the start that DYNIA has some attributes that makes the changes done with the sampling scheme of the GLUE method far from straightforward to replicate with DYNIA, but it is nevertheless considered by the author as an important area for further research in order to increase the practicability of the method. For this reason, an introduction to these concepts are included here, together with thoughts on the possibility to use them to enhance the efficiency of DYNIA.

Markov chain Monte Carlo is the name of algorithms that uses the previous sample value to randomly generate the next sample value, thus generating a Markov chain. For a Markov random variable, only information about the current state of the variable is needed to predict the future value. A Markov chain refers to a a sequence of random variables generated by a Markov process.

The basic idea of Markov chain Monte Carlo sampling is to sample from a probability distribution that based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. This can be used to generate "good" solutions with a higher probability because the samples are made from a distribution where "good" solutions have higher probabilities.

Genetic algorithms are search techniques used in computing to find solutions to optimization problems that serve as an alternative to traditional optimization techniques by using directed random searches to locate optimal solutions [28]. The search methods of genetic algorithms have their root in the mechanisms of evolution and natural genetics, and are as such a subclass of evolutionary algorithms. The techniques that are used are inspired by evolutionary traits such as inheritance, mutation, selection and crossover.

In evolutionary algorithms, the candidate solutions to the optimization problem play the role of individuals in a population. The search technique of the algorithm implements evolutionary traits that then are applied to generate a new generation of candidate solutions, i.e. a new population. The individual solution candidates of a population are evaluated with respect to a optimization function, thereby indicating how "strong" the individual is. The consequence of the evolutionary traits of the search algorithm is that the creation of the new generation of candidate solutions follow the natural phenomenon of "survival of the fittest", where individuals best suited to competition for resources and to a changing environment are most likely to survive

and reproduce.

Genetic algorithms rely on crossover, a mechanism of probabilistic and useful exchange of information among solutions, to locate better solutions. The probability of two individuals, "parents", in a population to produce an offspring is dependant on how well the parents perform with respect to the objective function, i.e. how "fit" they are. The new candidate solution, i.e. the offspring of the parents, is formed my inheritance and crossover of the genetic material of the parents. In the context of parameter estimation, where the "genes" of an individual consist of a model parameter set, this means that the offspring inherits a parameter set from the parents that is influenced by crossover of the parameter sets of the parents. In addition the parameter set of an offspring will be subject to a random mutation, which represent changes in the DNA sequence of a cell's genome during DNA replication.

Differential evolution (DE) is one genetic type of algorithm created for mathematical optimization of multidimensional functions that has good convergence properties to a global minimum [30]. The DE algorithm has demonstrated faster convergence with more certianty than many other acclaimed global optimization techniques, and ever more researchers are working with and on differential evolution. The DE algorithm requires few control variables, is robust and easy to use, and lends itself well to parallell computing [30].

Since a practical problem for the GLUE method is that the sample size has to be very large to achieve a reliable estimate of model uncertainties for models with a large number of parameters, there has been performed research on how to make this more effective. It has been shown that using an adaptive Markov Chain Monte Carlo algorithm improves the computational efficiency [5], the same with the use of a genetic algorithm [15]. These methods both reduce the necessary number of samples, and the sampled parameter sets they provide are all behavioural with respect to a certain threshold cost value. The reason why this is possible for GLUE is because this method is concerned with overall model output uncertainty, this means that the samples can be deemed behavioural or non-behavioural in the sampling and simulation process. For the DYNIA method however, these changes to the sampling scheme are not as intuitive to implement. Since the DYNIA method is concerned with how the parameter uncertainty is varying with time, it is not possible to decide if a sampled parameter set is behavioural or non-behavioural without performing the moving window analysis of the entire time series.

If these methods are to be applied to the DYNIA method, it seems like the only option is to perform the parameter sampling with respect to each time step, i.e. each window. In this case, a possible way of generating behavioural parameter sets (for each window) is to use a genetic algorithm such as DE. Multiple "threads" of the DE algorithm could be run. A selection, from all these runs of the algorithm, between the population members that exist after a specific number of generations, could be put together to obtain behavioural parameter sets. As mentioned, this will have to be done at each time step.

It is fairly obvious that a lot of more work has to be done before this theory could be implemented and tested. Unfortunately, it has not been possible to achieve this in this thesis, therefore it is rather proclaimed as an important topic for further research.

If it turns out that such an integration of a genetic algorithm effectively can be applied to the DYNIA method, it would also be interesting to investigate whether or not this approach to obtaining behaviour parameter samples improves efficiency with regards to the method for including DYNIA in a logging system that was discussed in Section 4.3

6 Conclusion and Further Work

This thesis has provided a thorough presentation of Dynamic Identifiability Analysis (DYNIA) and the concepts on which it is built upon, such as Generalized Likelihood Uncertainty Estimation and Regional Sensitivity Analysis. A goal of the thesis has been to create a text that can serve as a thorough introduction to the DYNIA method, in order to promote interest for the method with respect to system identification. The DYNIA method has not received attention in the scientific literature on system identification, and this text will hopefully raise awareness to the possibilities of the method. Another contribution of the thesis has been to show how the DYNIA method can be applied in the context of the work performed by Elgsæter [7], who suggested that research should be done to investigate how measures of uncertainty can be exploited to devise strategies for production optimization under uncertainty.

The possibility to be able to observe how the information content vary with time, and then using this to reevaluate process model parameters without the need to stop the production in order to perform experiments designed for the purpose of creating informative data, is an asset of the DYNIA method that could provide increased cost-effectiveness for many industrial systems, perhaps particularily in the oil and gas production industry.

Another main goal of the thesis was to implement the method, and to document and provide source code for this. As of this date, there does not exist any other information on how to implement the method, in fact there is generally little existing literature on the DYNIA method. The implementation will hopefully provide ground for testing and ideas that can lead to modifications that can improve the functionality of the method.

It has to be said that the DYNIA method has some limitations that are constraining the industrial applicability of the method and the implementation, at least in its current state. Exponential time complexity is obviously a major concern. Another concern is how selection of feasible parameter ranges, window size and tolerance limits for the behavioural parameter sets affect the results, this is an area for further research. These matters affects the reliability one can expect to have in the results, something that is especially problematic

if the DYNIA method is wanted to perform with little manual involvement.

The examples do however show promising results with respect to the basic functionality of DYNIA. The method does indicate periods of high and low information content, and the measurement of parameter identifiability also gave promising results in the examples.

One area for further research has already been mentioned, other important topics in this regard are mainly the modifications that were discussed in the previous chapter.

Parallel computation of the Monte Carlo sampling scheme and the moving window calculations were shown to have the possibility to yield good increases in efficiency. More research on this should be performed, and the theories should be applied to the implementation and testes.

Another important area is research on ways to improve the efficiency of the parameter sampling. Some ideas on possible modifications in this context have been given, and further research based on these ideas should be performed.

References

[1] T. Aalbers, S. Duane, R.-P. Kapsch, and A. Meghzifene. Measurement uncertainty - a practical guide for secondary standards dosimetry laboratories. Technical report, International Atomic Energy Agency, 2008.

[2] K. Beven. Generalised likelihood uncertainty estimation (glue). PUB-IAHS Workshop - Uncertainty Analysis in Environmental Modelling, 2004.

[3] H. Bieker, O. Slupphaug, and T. Johansen. Real time production optimization of offshore oil and gas production systems: A technology survey. *SPE Prduction & Operations*, 22(4):382 – 391, 2007. SPE 99446.

[4] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[5] R. Blasone, J. Vrugt, H. Madsen, D. Rosbjerg, B. Robinson, and G. Zyvoloski. Generalized likelihood uncertainty estimation (glue) using adaptive markov chain monte carlo sampling. *Advances in Water Resources*, 2008.

[6] K. P. Burnham and D. R. Anderson. *Model Selection and Inference*. Springer-Verlag, 2002.

[7] S. M. Elgsæter. *Modeling and Optimizing the Offshore Production of Oil and Gas under Uncertainty*. PhD thesis, Norwegian University of Science and Technology, 2008.

[8] P. Eykhoff. *System Identification*. Wiley, 1974.

[9] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT press, 1996.

[10] A. O. Fordal. Identifiability analysis of process data. Technical report, Norwegian University of Science and Technology, 2008.

[11] J. Freer, S. Beven, and B. Ambroise. Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the glue approach. *Water Resources Research, 32(7):2161-2713*, 1996.

[12] H. Harvey. Sensitivity analysis. http://www.floodrisknet.org.uk/glossary/sensitivity-analysis, 2006.

[13] G. Hornberger and R. Spear. An approach to the preliminary analysis of environmental systems. *Journal of Environmental Management, 12: 7-18*, 1981.

[14] O. Kahrs and W. Marquardt. The validity domain of hybrid models and its application in process optimization. *Chemical Engineering and Processing, 46: 1054-1066*, 2007.

[15] S. Khu and M. Werner. Reduction of monte carlo simulation runs for uncertainty estimation in hydrological modelling. *Hydrology & Earth System Sciences*, 2003.

[16] P. D. W. Kirk and M. P. H. Stumpf. Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics*, 25:1300 – 1306, 2009.

[17] V. Kumar. *Introduction to Parallel Computing*. Addison-Wesley Longman Publishing, 2002.

[18] L. Ljung. *System Identification*. Prentice-Hall, 1999.

[19] C. M. Macal, editor. *Model Verification and Validation*, 2005.

[20] J. Mikleš and M. Fikar. *Process Modelling, Identification and Control*. Springer-Verlag, 2007.

[21] J. B. Moore. Persistence of excitation in extended least squares. *IEEE Transactions on Automatic Control, 28: 60*, 1983.

[22] M. Nikravesh, M. Soroush, and R. M. Johnston, editors. *Nonlinear Control of an Oil Well*, 1997. Proceedings of the American Control Conference.

[23] S. Park and D. Himmelblau. Parameter estimation and unique identifiability. *The Chemical Engineering Journal, 25: 163-174*, 1982.

[24] P. Pascual, N. Stiber, and E. Sunderland, editors. *Draft Guidance on the Development, Evaluation, and Application of Regulatory Envrionmental Models*, 2003.

[25] J. Rosenthal. Parallel computing and monte carlo algorithms. *Far East Journal of Theoretical Statistics*, 2000.

[26] P. Sayers, B. Gouldby, J. Simm, I. Meadowcroft, and J. Hall. Risk, performance and uncertainty in flood and coastal defence - a review. Technical report, DEFRA/Environment agency - Flood and Coastal Defence R & D Programme, Wallingford, 2002.

[27] R. Spear and G. Hornberger. Eutrophication in peel inlet, ii, identification of critical uncertainties via general sensitivity analysis. *Water Resources Research, 14: 43-49*, 1980.

[28] M. Srinivas and L. M. Patnaik. Genetic algorithms: A survey. *Computer*, 1994.

[29] R. Stedinger, R. Vogel, S. Lee, and R. Batchelder. Appraisal of the generalised likelihood uncertainty estimation (glue) method. *Water Resources Research, 44: 17pp.*, 2008.

[30] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997.

[31] T. Wagener, L. Camacho, and H. Wheater. Dynamic identifiability analysis of the transient storage model for solute transport in rivers. *Journal of Hydroinformatics, 4(3): 199-211*, 2002.

[32] T. Wagener, N. McIntyre, M. Lees, H. Wheater, and H. Gupta. Towards reduced uncertainty in conceptual rainfall-runoff modelling: Dynamic identifiability analysis. *Hydrological Processes, 17: 455-476*, 2003.

[33] T. Wagener and H. Wheater. *Mone-Carlo Analysis Toolbox User Manual - Version 5*, 2004.

[34] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier Inc., 2005.

Appendix A: A Bayesian Approach to Identifiability

This appendix is based on information in [10]

A sample-based uncertainty analysis can be linked to a concept that is better known in the traditional discussion on system identification; the *maximum a posteriori (MAP) estimate*. The MAP estimate uses a Bayesian approach to the parameter estimation problem by considering the parameter set θ as a random variable with a certain a priori probability distribution function. Let us first define the a priori PDF as

$$g_\theta(\theta^*) = P(\theta = \theta^*), \tag{1}$$

which give the probability of θ being equal to θ^*.

Then, let us define a *likelihood* function as

$$\theta \to f(y_m|\theta), \tag{2}$$

this is a conditional PDF for the system output that give the probability of the model output y_m being equal to the system output y, given that the model use the parameter set θ.

Using Bayes' theorem we can use these PDFs to find the posterior conditional PDF for θ:

$$h(\theta) = P(\theta|y) = \frac{f(y_m|\theta) \cdot g_\theta(\theta)}{\int_\theta f(y|\theta')g_\theta(\theta')d\theta'} \tag{3}$$

$h(\theta)$ give a distribution of the likelihood for the different parameter values. The MAP estimate is the parameter value that maximize $h(\theta)$.

By comparing this to the description of the sample-based uncertainty analysis in Section 1.3, we see that the uncertainty analysis give us all the information we need to calculate the probability/likelihood distribution $h(\theta)$. This means that by performing an uncertainty analysis, we can acquire information on uncertainty, sensitivity and likelihood distribution of the parameter values.

Appendix B: Matlab Code

Matlab code for the file that runs the first example, and for the files that are the "meat" of the implementation of DYNIA.

example_firstorder.m:

```
clc;
close all;
clear all;

size = 500;

time = 1:size;

u = 35*ones(size,1);

u(140:160) = 10; u(330:332) = 10;
u(400:size) = 0;

b1 = 2;
a1 = 3;
delay = 0;

sys = tf(b1, [a1 1], 'InputDelay', delay);
sys = ss(sys);
[a,b,c,d] = ssdata(sys);

y0 = 0;

[y, t] = lsim(sys, u, time, y0/c);

noiselev = 0.5;
yn = y + randn(length(y),1)*noiselev;

figure
plot(time, u, time, yn)
```

```
% Model structure: 1st order. No time delay. y/u (s) = K/(T_1 + s)

% MC options
a1_lb    = 0;  % time constant lower [minutes]
a1_up    = 7;  % time constant upper [minutes]
b1_lb   = 0;  % gain lower
b1_up   = 7;   % gain upper

% delay_lb = 80;
% delay_up = 120;

ns      = 2000;  % no. of MC sims

% generate matrix of random samples from parameter distributions
a1_mc=(rand(ns,1)*(a1_up-a1_lb))+a1_lb;
b1_mc=(rand(ns,1)*(b1_up-b1_lb))+b1_lb;
% delay_mc=(rand(ns,1)*(delay_up-delay_lb))+delay_lb;



% intialise output matrices
sse=zeros(ns,1);abias=zeros(ns,1);peako=zeros(ns,1);
mct=zeros(ns,size);

h = waitbar(0,'Running Monte-Carlo simulation, please wait...');


for k=1:ns
   % First order
   sys = tf(b1_mc(k),[a1_mc(k) 1], 'InputDelay', delay);
   sys = ss(sys);
   [a,b,c,d] = ssdata(sys);
   x = 70;

   % Euler method
   for(i = 1:size)
       x_d = a*x + b*u(i);
       x = x + x_d; %time step = 1
       mct(k,i) = c*x;
   end
```

```
   % Alternative simulation method (longer running time)
   %[mct(k,:), t_mc] = lsim(sys,u, 1:size);

   e=y-mct(k,:)';
   sse(k)=sum(e.^2); % sum of squared errors
   %abias(k)=abs(sum(e)/length(mct(k,:))); % absolute bias
   %peako(k)=max(mct(k,:));
   waitbar(k/ns)
end
close(h)


%% RUN DYNIA - MOVING WINDOW

bins = 40;          % # of bins for gradient calculation of CLD
percentage = 10;    % percentage to be included in behavioural set
n = 10;             % window size = 2*n + 1

for i=1:2

    selectedPar = i;

    if selectedPar == 1
        par = a1_mc;
        lb = a1_lb;
        ub = a1_up;
        partitle = 'Time Constant1';
    end

    if selectedPar == 2
        par = b1_mc;
        lb = b1_lb;
        ub = b1_up;
        partitle = 'Gain1';
    end

%     if selectedPar == 3
%         par = delay_mc;
%         lb = delay_lb;
%         ub = delay_up;
%         partitle = 'Time Delay';
%     end

    [IC, CI, gradient_data] = movingWindowCalcs(size, par, lb, ub, ns, mct, y, bins,
```

```
    %% Information content plot
    figure
    %plot(t, IC(t));
    plot(time, IC);

    xlabel('Time Step')
    ylabel('Information Content')
    title(['Information content of parameter: ', partitle])
    ylim([0 1])

    %% Parameter estimation plot
    figure
    plot(time, CI(:, 1), time, CI(:, 2))

    hold on

    colormap(flipud(gray))
    x_range = time;
    y_step = (ub-lb)/(bins-1);
    y_range = lb :y_step:ub;

    pcolor(x_range, y_range, gradient_data)
    shading flat
    axis([time(1), time(size), lb, ub])

    xlabel('Time Step')
    ylabel('Parameter Value')
    title(['Uncertainty band and CLD gradient of parameter: ', partitle])

    hold off

end
```

movingWindowCalcs.m

```
%% Moving-Window Calculations. Dynamic employment of the DYNIA method.

function [IC, CI_t, gradient_data] = movingWindowCalcs(TIME, par, lb, ub, ns, mct, y

% Window size
wSize = 2*n + 1;

% Set up vectors for storage
IC = zeros(TIME,1);
CI_t = zeros(TIME, 2);
gradient_data = zeros(bins, TIME);

best = zeros(TIME,ns*(percentage/100),2);
tStep = 1;

% Output which parameter is being analyzed
h = waitbar(0,['Performing Moving-Window calculations for parameter: ', partitle,' .

% Perform moving-window calculations for each time step
while (tStep <= TIME)

    wPar = par;             % parameter to be analyzed
    wMAE = zeros(ns,1);     % mean absolute error

    % Calculate MAE for the time steps before full window size is possible
    if(tStep <= n)
        for k = 1:ns
            wMAE(k) = (sum(abs( yn(1 : tStep+n) - mct(k, 1 : tStep+n)' )))/(tStep+n)
        end
    end

    % Calculate MAE for time steps where full window size is possible
    if (tStep > n && tStep <= (TIME - n))
        for k = 1:ns
             wMAE(k) = (sum(abs( yn(tStep-n : tStep+n) - mct(k, tStep-n : tStep+n)' )
        end
    end

    % Calculate MAE for time steps after full window size is possible
    if (tStep > (TIME - n))
        for k = 1:ns
            wMAE(k) = (sum(abs( yn(tStep-n : TIME) - mct(k, tStep-n : TIME)')))/(TIM
```

```
        end
    end

    wCost = wMAE.^2;    % SSE value for the window


    % Select top percentage of MC parameter sets
    count_max = floor(length(wCost) * (percentage/100));

    %best_mat column 1: cost values, column 2: corresponding parameter values
    [best_mat, best_indices] = selectBestParameters(wCost, wPar, count_max);

    % Save the behavioural set for each time step
    best(tStep,:,1) = best_mat(:,1);
    best(tStep,:,2) = best_mat(:,2);

    % Run the DYNIA algorithm for the current window
    [lhood_cumulated CI lhood, gradient_vec] = runDynia(ns, par, best_mat, lb, ub, bi

    % Save confidence interval for each time step
    CI_t(tStep, 1) = CI(1);
    CI_t(tStep, 2) = CI(2);

    % Calculate and save information content measure for each time step
    IC(tStep) = 1 - (CI(2) - CI(1))/(ub - lb);


    %% Identifiability

    % Save gradient data for each time step
    gradient_data(:,tStep) = gradient_vec(:,1);


  %% Go to next time step
  waitbar(tStep/TIME);

  tStep = tStep + 1;

end
close(h)
```

runDynia.m:

```
%% Run the DYNIA analysis

function [lhood_cumulated, CI, lhood, gradient_vec] =  runDynia(ns, par, best_mat, l

size = length(best_mat(:,1));

%% Calculate likelihood measure for each simulated parameter set.

lhood = calcLikelihood(best_mat);


%% Cumulative likelihood distribution

[par_sorted, par_sorted_indices] = sort(best_mat(:, 2));
[all_par_sorted, all_par_sorted_indices] = sort(par);

[lhood_cumulated, lhood_c_all] = cumulLikelihood(ns, all_par_sorted, par_sorted, par

%% Gradient of cumulative likelihood distribution

gradient_vec = cld_gradient(bins, ub, lb, par_sorted, lhood_cumulated, size);


%% Calculate confidence interval for cumulative likelihood distribution

conf_level = 90;
CI = confLimits(par_sorted, size, conf_level);
```

calcLikelihood.m:

```
%% Calculation of likelihood measure

function [lhood] = calcLikelihood(best_mat)

size = length(best_mat);
lhood = zeros(1, size);

% Algorithm for calculation of the likelihood measure
for it = 1:size
    lhood(it) = 1 - best_mat(it, 1);
```

```
end

if( min(lhood) < 0)
    lhood(:) = lhood(:) - min(lhood);
end

lhood(:) = lhood(:)/sum(lhood);
```

cumulLikelihood.m:

```
%% Calculation of cumulative likelihood distribution (CLD)

function [lhood_cumulated lhood_c_all] = cumulLikelihood(ns, all_par_sorted, par_sor

size = length(par_sorted_indices);

lhood_cumulated = zeros(1, size);

% Calculate CLD for each sampled (behaviour) value of the parameter being analyzed
lhood_cumulated(1) = lhood(par_sorted_indices(1));
for it = 2:size
    lhood_cumulated(it) = lhood_cumulated(it-1) + lhood(par_sorted_indices(it));
end

% CLD value is set to zero and one for parameter values lower, respectively
% higher than the lowest, respectively highest parameter value in the behavioural se
it=1;
while(all_par_sorted(it)<par_sorted(1))
    it = it+1;
end

lhood_c_all(1 : (it-1) ) = 0;
count=0;
while( (all_par_sorted(it)>=par_sorted(1)) && (all_par_sorted(it) < par_sorted(size)
    count = count + 1;
    it = it + 1;
end

lhood_c_all(it : it+size-1) = lhood_cumulated(:);
lhood_c_all(it+size : ns-count) = 1;
```

cld_gradient.m:

```
%% Calculation of gradients of a cumulative likelihood distribution

function gradient_vec = cld_gradient(bins, ub, lb, par_sorted, lhood_cumulated, size)

gradient_vec = zeros(bins, 4);

next_limit = lb + (ub-lb)/(bins-1);
prev_limit = next_limit;

bin = 1;

% Gradient is zero for parameter values below the lowest value in the
% behavioural set (since the CLD is constant for these values)
while ( (next_limit < par_sorted(1)) && (next_limit <= ub) && (bin<=bins))
    next_limit = next_limit + (ub-lb)/(bins-1);

    gradient_vec(bin,1) = 0;
    gradient_vec(bin,2) = prev_limit;
    gradient_vec(bin,3) = (prev_limit + next_limit)/2;
    gradient_vec(bin,4) = next_limit;
    bin = bin + 1;
    prev_limit = next_limit;
end

% Gradient for a bin (of parameter values) is calculated by subracting the CLD value
% beginning of the bin from the CLD value at the end of the bin.

start_index = 1;
while( (next_limit <= par_sorted(size)) && (next_limit <= ub) && (bin <= bins) )
    index = start_index;
    while ( (par_sorted(index) < next_limit) && (index <= size-1) )
        index = index + 1;
    end
    end_index = index-1;
    next_limit = next_limit + (ub-lb)/(bins-1);

    gradient_vec(bin,1) = lhood_cumulated(end_index) - lhood_cumulated(start_index);
    gradient_vec(bin,2) = prev_limit;
    gradient_vec(bin,3) = (prev_limit + next_limit)/2;
    gradient_vec(bin,4) = next_limit;
    bin = bin+1;
    prev_limit = next_limit;
```

```
    start_index = end_index + 1;
end


% Gradient is zero for parameter values above the highest value in the
% behavioural set (since the CLD is constant for these values)
while( (next_limit > par_sorted(size)) && (next_limit <= ub) && (bin <= bins))
    next_limit = next_limit + (ub-lb)/(bins-1);

    gradient_vec(bin,1) = 0;
    gradient_vec(bin,2) = prev_limit;
    gradient_vec(bin,3) = (prev_limit + next_limit)/2;
    gradient_vec(bin,4) = next_limit;

    prev_limit = next_limit;
    bin = bin+1;
end
```

confLimits.m:

```
%% Calculate confidence limits of the CLD

function [CI] = confLimits(par_sorted, count_max, conf_level)
c1 = (1 - conf_level/100)/2;
c2 = 1 - c1;

cl = [c1 c2];                           % confidence levels
cli = floor(cl*(count_max - 1)) + 1;    % confidence interval indexes

CI = par_sorted(cli);
```