Roar Nybø

Efficient Drilling Problem Detection

– Early fault detection by the combination of physical models and artificial intelligence

Thesis for the degree of Philosophiae Doctor

Trondheim, September 2009

Norwegian University of Science and Technology
Faculty of Engineering Science and Technology
Department of Petroleum Engineering and Applied Geophysics

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering Science and Technology
Department of Petroleum Engineering and Applied Geophysics

978-82-471-1097-3 (printed ver.)
978-82-471-1098-0 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2009:170

Printed by NTNU-trykk

Preface

This synthesis and a collection of papers are submitted for the degree of philosophiae doctor (PhD) at the Department of Petroleum Engineering and Applied Geophysics, Norwegian University of Science and Technology.
The dissertation is divided into two parts. One part introducing the theory and detailing our findings and one part consisting of four papers:

Paper I NYBØ, R., BJØRKEVOLL, K. S. & ROMMETVEIT, R. (2008) SPE 112212-MS, Spotting a False Alarm—Integrating Experience and Real-Time Analysis With Artificial Intelligence. *Intelligent Energy Conference and Exhibition.* Amsterdam, The Netherlands, Society of Petroleum Engineers.

Paper II NYBØ, R., BJØRKEVOLL, K. S., ROMMETVEIT, R., SKALLE, P. & HERBERT, M. (2008) SPE 113776 - Improved And Robust Drilling Simulators Using Past Real-Time Measurements And Artificial Intelligence. *2008 SPE Europec/EAGE Annual Conference and Exhibition.* Rome, Italy.

Paper III NYBØ, R. (2008) Time series opportunities in the petroleum industry. *ESTSP 08, European Symposium on Time Series Prediction, Porvoo, Finland.* An extended version of the paper presented here has also been submitted to an ESTSP 08 special issue of the journal Neurocomputing. (Elsevier)

Paper IV GULSRUD, T. O., BJØRKEVOLL, K. S. & NYBØ, R. (2009) Statistical Method For Detection Of Poor Hole Cleaning And Stuck Pipe. *Offshore Europe 2009, Aberdeen, UK.*

This work has been funded by The Research Council of Norway (project number 169439/S30) and partly by the Center for Integrated Operations in the Petroleum Industry.
The work has been carried out while employed at SINTEF Petroleum Research

Abstract

The drilling of an oil or gas well is an expensive undertaking. Hence, it is not surprising that mistakes and accidents during drilling incur a high cost. Accidents could result in the loss of expensive equipment and subsequent delays setting back the operation for days or weeks and thus running up large bills on rig-time and personnel hours. Some types of accidents also pose a risk to the personnel or the environment. In this dissertation we study alarm systems which could give the driller an early warning of upcoming problems, and thus provide time to avoid these accidents. We explore alarm systems which combine advanced physical models of the well and drilling process with artificial intelligence and time series analysis. Finally, we determine the advantages as well as the challenges of this approach.

It is our hope that this dissertation is accessible to both practitioners in machine learning and control engineering, as well as to petroleum engineers with a passing familiarity with machine learning. Hence this dissertation starts with a quick introduction to drilling problems and some terms from time series analysis and machine learning. We then briefly describe the theory of observer-based fault detection and isolation. Theories of supervisory control systems are also introduced, as these concern both the choice of algorithms and how AI-based alarm systems integrate with the rest of the operation. From chapter 6 and onward, the challenges to fault detection in drilling are discussed. We focus on clarifying what restrictions the available training data put on our choice of machine learning methods. In chapter 8 and 9, we propose ways to combine machine learning and observer-based fault detection. Experimental results are presented in chapter 10, before we end with concluding remarks in chapter 11.

Our main conclusion, reflected in our experimental results, is that physical models and artificial intelligence can be combined to produce hybrid alarm systems that are better than what we could have achieved using these approaches separately.
When using artificial intelligence we treat fault detection in drilling as a machine learning problem. In the course of our work we find that this problem domain differs in important respects from textbook examples of machine learning problems. Determining the distinctive characteristics of this problem domain is crucial in designing the alarm system. Drawing on examples from different fields we determine these characteristics and propose novel alarm system architectures that build on recent developments in machine learning.

Acknowledgements

The work presented in this thesis/dissertation is the result of three years work funded by The Research Council of Norway.

I would like to take this opportunity to express my gratitude to some people that has been of great help during this period. I would first like to thank my supervisors, Knut S. Bjørkevoll at SINTEF Petroleum Research and Pål Skalle at the Department of Petroleum Engineering and Applied Geophysics, NTNU. I am grateful for their support and for giving me the opportunity to explore such an exciting dissertation topic.

My time has been spent mainly at SINTEF Petroleum Research in Bergen, where I have had the pleasure to be part of the Department of Drilling and Well Construction. I'm grateful towards my drilling colleagues in Bergen and Stavanger, for teaching me about drilling, for interesting discussions and for making SINTEF a nice place to work. I would also like to thank ConocoPhillips and StatoilHydro for sharing their time-series data with us. These data, from their past drilling operations, formed a basis for this dissertation.

Last but not least I would like to extend my thanks to my friends and family, and especially to my lovely wife Irlin, for being my fixed point of reference.

1 Introduction

1.1 Today's challenges

Drilling for oil and gas is a task with a high risk of costly accidents. One underlying cause is noise and uncertainty in the available information. For instance, a drilling operation proceeds through different layers of rock which each may require different strategies by the driller, but the exact properties and positions of the layers may not be known. Feedback from the instruments is also fraught with uncertainty.
Measurements such as bottom hole pressure, mud flow, and hook load are noisy and influenced by a large number of effects. Still, experts analysing time series recorded prior to a fault, often concede that the signs of an imminent fault were in fact visible and that many of the accidents were avoidable mistakes.
This leads us to a second underlying cause of faults, real-time information overload. Much work has gone into producing software that either condenses the real-time data into a more manageable form for the driller, or produces an alarm when the system is getting near a fault. The most advanced alarm systems today model the well and the drilling process. These models let the driller anticipate the effects of his actions and help the alarm systems separate normal behaviour from imminent faults. After presenting the necessary background knowledge, we go into more detail on the state of the art in chapter 7.

1.2 Our objective

The objective of this dissertation is to improve existing alarm systems by incorporating both machine learning and physical models.
Frequent interruptions by false alarms have been recognized as a major problem for drillers in the North Sea (Heber and Åsland, 2007). This erodes trust in the alarm system, risking that correct alarms go unheeded. Increasing the sensitivity of an alarm system tends to increase both the detection rate and the number of false alarms. When the false alarm rate is already too high, this in effect puts a cap on the sensitivity, meaning that existing alarm systems are not run at their full potential. Finally, false alarms distract the driller from his or her job, making it more likely that the driller makes a mistake or overlooks signs of real problems. A priority in this dissertation has thus been to find ways to reduce the false alarm rate.

In addition to the main goal of better alarm systems, it has also been an objective to analyse artificial intelligence (AI) and hybrid alarm systems as parts of the larger supervisory systems in petroleum production and Integrated Operations, where information and communication technology enable new work processes in the industry (Epsis and ABB, 2006).

1.3 Methods

In this dissertation we make use of a previously developed physical model of the wellbore and drilling process. This model is part of the eDrilling system which has been developed by SINTEF and co-operators (Petersen et al., 2006).
The eDrilling system runs in real-time during a drilling operation, taking real-time measurements as input and predicting downhole conditions. This includes pressure, temperature, and mass transport along the wellpath. The model calculates factors such

as pump rate, movement of the drill string and choice of drilling mud affect the well. The eDrilling system also presents a 3D visualization of the well and drilling process. It is also possible to feed the eDrilling system with time series recorded during past drilling operations. We may then re-run the operation in fast-forward to analyse the drilling operation and events therein.

Several time series recorded during the drilling of wells in the North Sea have been available to us. Some of these contained drilling problems or faults. Our method of investigation has been to study these time series for early signs of drilling problems and re-run time series through eDrilling to study how the model behaves during a fault. The time series together with output from the eDrilling model constitutes a data set upon which we have tested different machine learning methods.

This dissertation makes extensive use of machine learning methods, a field which tends to overlap with topics such as soft computing, AI, and data mining. Without any opinion on the delineation between these subfields, an “AI” in this dissertation is simply meant to indicate an instance of a predictor or a classifier. This includes some simple and transparent methods that are not normally referred to as AIs, but nevertheless share many of the same challenges.

To address the Integrated Operations perspective and how our work is relevant to the larger petroleum industry, we have discussed the AI alarm system as part of a larger integrated system of supervision, control and optimization. Such systems are central to many IO initiatives and the discussion of these systems further informs our alarm system design.

2 Drilling problems

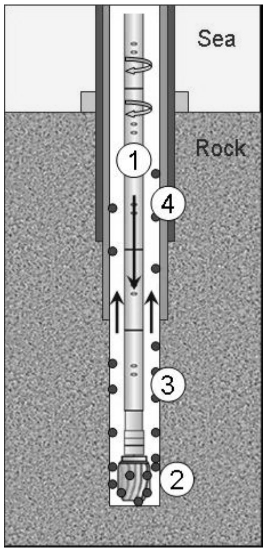

Figure 1: Simplified drilling schematics

The basic offshore drilling operation can, with reference to Figure 1, be described as follows: A rotating pipe (1) extends from the rig to the bottom of the well, where a bottom hole assembly including a drill bit (2) is mounted. The drillbit crushes the rock into *cuttings*. At the same time drilling mud is being pumped down the pipe. The mud returns to the rig through the *annulus*, the space between the pipe and the wall of the borehole. The mud carries the rock cuttings (3) along with it, up to the rig. As drilling progresses, the wall of the well is periodically fitted with a protective casing (4). To replace worn-out drill bits, it is necessary to pull the pipe out of the hole. This and the subsequent reinsertion is called *tripping*.

In this dissertation, we will focus on two complications that may arise during this operation. Under certain conditions, such as when drilling into a new geological formation, the pore pressure of the formation may exceed well pressure and gas or fluid may flow into the well. This displaces the mud, which leads to a larger mud return rate at the rig.
Extra mud return is the most significant sign of a *kick* (Watson et al., 2003). Bubbles of gas will expand as they rise, so that the amount of fluid displaced is not proportional to the original influx. In the case of gas, or when the density of the formation fluid is less than that of the mud, an influx will cause a pressure reduction in the well, which further destabilizes the situation. This pressure loss is another early indicator of a kick[1]. Acoustic methods provide indications of a gas kick but they are unsuitable for the earliest detection and unsuitable for deep and high pressure high temperature wells (HPHT), because dissolved gas is harder to detect than free bubbles (Watson et al., 2003).

The kick proceeds on a timescale of seconds and minutes and the drilling crew must act swiftly to prevent what could in the worst case become a full-scale *blow-out*. Early detection requires both reliable flow measurements and the ability to predict harmless changes in the fluid flow, such as fluid displaced by the drill string during tripping. Otherwise, these effects will result in false kick alarms.
The e-drilling software (Petersen et al., 2006) calculates many but not all of these harmless effects. Of special significance in this dissertation is *pipe draining*: When the pumps are stopped, the flow of mud out of the well stops abruptly. However, it takes a few minutes for the pipes between the well and the flow meter to empty. This effect is not included in the e-drilling system at present and the meter readings could therefore be misinterpreted as fluid displaced by a kick.

[1] In the case of a slim hole, the extra annulus friction pressure, caused by the influx flow, may be higher than the reduction in hydrostatic pressure, so that standpipe pressure actually increases.

Successful drilling also requires that cuttings are transported to the surface. If not, we risk that the cuttings settles out and eventually packs around the drill string. This is termed *poor hole cleaning* and may result in a *stuck pipe* incident. A worst-case scenario is that the drill string breaks, resulting in the loss of the bottom hole assembly. In such a case, much of the well will be blocked by the lost pipe. It will be required to either drill long side-tracks or perform time-consuming *fishing operations* to clear the hole. Poor hole cleaning is a condition that often builds up over time, thus the signature of the impending fault should in theory be visible long before the fault. The signatures are however less clear-cut than for kicks. Signatures include:

- An erratic torque on the drill string. There could be several factors influencing torque, for instance that the string is repeatedly getting stuck in the cuttings, wound up and spun free.
- An increase in bottom hole pressure that is otherwise unaccounted for, may indicate a tight spot with cuttings packings causing flow restrictions further up the annulus
- An unexpected hook load. The hook load is the tensional force of the drill string exerted as it is suspended from the rig. The drill string may partially rest on a tight packing of cuttings while running into the well, causing the hook load to be lower than anticipated.

A stuck pipe may also be caused by differential sticking or by a borehole which is producing cavings or which is collapsing. These processes share some but not all of the characteristics of poor hole cleaning. We refer to (Aldred et al., 1999) for an introduction. As in the case of a kick, we compare the measurements with predictions of the fault-free case, but for stuck pipe we need to focus more on defining a reliable and robust fault signature.
For both kick and stuck pipe, in particular, there are strong indications of the fault in one or two variables and weaker correlated signals in other variables, which we may or may not be able to exploit.

3 Some machine learning concepts

In this chapter we briefly introduce some concepts and theories from machine learning, which are central to discussions later in this dissertation.

3.1 The curse of dimensionality

Both AI methods, such as neural networks, and simpler methods such as linear regression, perform a function approximation. Given a set of input examples, each with dimensionality d and corresponding outputs, a function is found which reproduces the examples fairly well and manage to predict the output belonging to new examples. We know that more examples generally lead to a better approximation of the function, but how this depends on dimensionality is perhaps not intuitive. If we make no assumptions about the form of the function, a function of d variables need n examples to get the approximation down to an error of ε, where (Verleysen, 2003):

$$n = (1/\varepsilon)^d \tag{3.1}$$

More precisely, n is the number of evenly spaced points[2] that need to be sampled in a d - dimensional hypercube with sides of length 1 so that the distance between the points is as low as ε. This means that the number of examples needed increases exponentially with the dimensionality of the input. It has the at first counter-intuitive implication that more information may lead to worse performance.

The problem is mirrored in data-mining approaches that produce a hypothesis about the data. In a seminal paper by Ioannidis (Ioannidis, 2005) the case of analysing medical data was discussed. n patients are screened for a large number of variables, such as genes or protein expression. Here the dimensionality d easily exceeds several thousands and the output to be determined is whether the patient does or does not have a disease. The goal is to find a statistically significant correlation between a gene and the disease. Standard techniques have a small chance of reporting an unrelated gene as linked to the disease. This is known as a type I error. Ioannidis showed in a convincing fashion that when this small probability of error is repeated for thousands of genes, type I errors could easily outnumber the true findings. In statistical and machine learning terms, this is a bad case of over-fitting or over-learning.

The curse of dimensionality demands that we enforce some best practices when dealing with high-dimensional data. Dimensionality reduction methods need to be deployed and our feature selection should not include parameters that we do not initially believe have a bearing on the output. Neither should we work with no assumptions on the function we try to approximate, but use functions on a form that is likely to reflect the problem at hand. That is, we need to employ a-priori information.

3.2 A-priori information

The curse of dimensionality makes a strong case for including a priori information in the pre-processing and analysis. This sentiment is mirrored in several other well-known results from machine learning and computer science. *The no free lunch*

[2] In reality, we may not be able to obtain evenly spaced points

theorem (Wolpert, 1996) , makes it clear that no method for optimizing our function is better than any other, if no assumptions are made about the underlying true function. The *bias-variance dilemma* goes into more detail on the performance:

Given a training data set with n examples of inputs and output $D = \{(y_i, \bar{x}_i),\ i = 1, 2, \ldots n\}$, we generate a function approximation $g(\bar{x}, D)$ which depends on the training set. The function may perform good or bad on new examples (the test set) and the bias-variance dilemma discusses how well the function performs compared to the MSE optimal regressor denoted by $E[y \mid \bar{x}]$. Our function's mean square deviation from the optimal (Theodoridis and Koutroumbas, 2006) is given by:

$$E_D\left[\left(g(\bar{x},D) - E[y \mid \bar{x}]\right)^2\right] = \left(E_D\left[g(\bar{x},D)\right] - E[y \mid \bar{x}]\right)^2 + E_D\left[\left(g(\bar{x},D) - E_D\left[g(\bar{x},D)\right]\right)^2\right] \tag{3.2}$$

We see that the first right hand side term can be identified as the bias, a tendency of our function to settle at some configuration different from the optimal. The second term is the variance of our function, how much it is prone to vary depending on the training set. For a finite number of training examples the dilemma states that decreasing the bias increases the variance and vice versa. For instance, a complicated model may be tuned and tweaked to fit the training set near perfectly, thus achieving a low bias, but it will show a high variance on new data. (Refer to Figure 2.) The use of a priori knowledge to restrict the forms the function can take, may however reduce both bias and variance at the same time (Theodoridis and Koutroumbas, 2006).

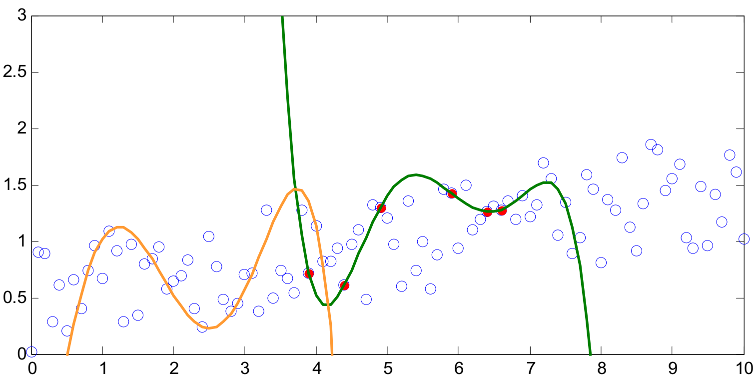

Figure 2: A fifth degree polynomial (green) is fitted to a training set of six examples (red) drawn from a larger population (blue). The polynomial has low bias in the training set but performs badly on new examples. A polynomial trained on six different examples (orange) illustrates the large variance of polynomials trained on different training sets.

In Figure 2 it is obvious that a less complex function, like a first degree polynomial, would have generalised better to new examples given the same training set. This notion of *generalisation performance* is made explicit by the *VC-dimension* (Theodoridis and Koutroumbas, 2006). Simply put, the VC-dimension is the "capacity" of our AI or function, which in the case of neural networks is roughly the number of free parameters to be fixed during training. Good generalisation performance can only be expected if the number of examples well exceed the VC-dimension.

While the no free lunch theorem instructs us that statements like “Method A is better than method B” are false unless specified for a specific problem domain, the bias-variance dilemma tells us to choose methods based on expert knowledge about the system. Both the latter and the VC-dimension further recommend an occhams razor approach, where the function or AI should be kept as simple as possible.

There are many approaches in the literature to formulating and implementing a-priori knowledge. Constraining the allowed parameters (Abonyi et al., 2000) (Lauer and Bloch, 2008), initializing the model with a-priori information before tuning it with training examples (Zhao and Dillon, 1997) and modifying the cost function (Papathanassiou and Petrou, 2002) are all possible approaches.
Some pre-processing is also concerned with this. While smoothing a time-series may be necessary for some algorithms to work, it is also a sort of dimensionality reduction, using a-priori knowledge that useful information is not found in the high-frequency range. De-trending or removing seasonality in a time-series may also be said to introduce a-priori information and more so employ a simple model of the system, predicting part of the raw data. Combined with an AI that has no a-priori information, this qualifies as a grey-box approach.

3.3 Black-, white- and grey-box models

Black-box models are models which can not be easily examined or where the system we try to model is itself a black box. That is, we have knowledge of how it responds to input, but not of its components or internal dynamics. In this case, all we ask of our model is that it reproduces the behaviour of the system, not that its internal workings correspond to that of the real system in any way.

The term *grey-box modelling* refers to an approach that lies in between white and black box approaches. Examples include neural networks that are fed not only the raw data but combinations of the parameters that are known to be relevant, such as the Reynolds number in a pipe flow problem. Another grey-box approach is to construct a simple physical model where its parameters are determined by the historical data (Awasthi et al., 2008) or a model where only some components are black boxes. In chapter 8.1 we discuss one such approach in more detail.

3.4 Ensemble methods

Ensemble learning is a collection of methods in machine learning which utilize several AIs simultaneously. The aim is to achieve a better performance than each AI alone or at least avoid the worst-case performance of a single AI. We will here introduce one class of methods called *bagging*.
An acronym for bootstrap aggregating, (Breiman, 1996a) bagging is a method where a set of AI classifiers or predictors are presented with the same problem and the ensemble output is arrived at by a form of voting. Two simple yet effective choices of voting are a majority vote for classifiers and averaging for predictors.
For bagging to work, the AIs need to be diverse in their predictions. If all the AIs produce the same output, the result of bagging would be no different from that of the

individual AI. This diversity can be accomplished e.g. by using different types of AIs, using different initial configurations of the AIs or choosing different subsets of input variables (Polikar, 2008) such as the random subspace method by (Ho, 1998). But the most common way to achieve diversity is to train similar AIs using different subsets of the available training data. Promoting diversity this way works best for AIs that are "unstable" in the sense that a small change in their training set can have a large effect on the final configuration of the trained AI. Many AIs such as neural networks and classification trees are unstable in this sense, with the k-means algorithm being a notable exception (Breiman, 1996b). Ensemble learning has been proposed as a solution for many data analysis challenges, such as data fusion and incremental learning. For an overview, see (Polikar, 2007, Valentini, 2003).

Ensemble methods are intuitively appealing, but their reliability and working mechanisms have been the subject of numerous studies. It was established by (Hansen and Salamon, 1990) that bagging reduces the variance of the prediction. However, the full picture has since been a matter of debate, with empirical studies yielding contradicting results. In theoretical studies, (Friedman and Hall, 2007) proceeded by decomposing smooth estimators into linear and higher-order terms and treated bagging using training set resampling. They found that variance reduction improved the higher order terms, with the linear term remaining unaffected. (Buja and Stuetzle, 2000) built on this to identify effects on variance, bias and mean square error. Cases were identified where bagging could in fact increase variance. It was also shown that (squared) bias always increases during bagging and that the detrimental effect of bias could outweigh variance reduction in some cases, typically for small resample sizes. (Buhlmann and Yu, 2002) analyzed the case of nonsmooth unstable predictors such as decision trees and found that in this case the first order term could be substantially improved. It is not clear if these results contradict the claim by (Kong and Dietterich, 1995) that some methods may reduce both bias and variance simultaneously, or if it the different claims are for different ensemble architectures.
While these approaches focus on the variance, the performance of ensemble systems has also been explained in terms of the classification margin, drawing on the theory of large margin classifiers such as support vector machines. This explanation has been especially convincing for boosting methods, but is also used to explain the effect of bagging (Schapire et al., 1998). It was shown in (Domingos, 2000) that margin can be expressed in terms of certain notions of variance and bias, allowing a unified treatment.

4 Theories of supervision and control

To understand the role of an alarm system, it is necessary to view it as part of a larger supervisory system that acts on the alarms. We present the established theory on this subject, first by a relevant example, then with an elaboration on the underlying structures as presented in the literature. When the drilling alarm system is put into this context, it becomes clear what data the system must work with and what forms the output can reasonably be permitted to take. This does in turn narrows the choice of machine learning methods. The supervisory system also serves as a roadmap for future integration and hybridization of machine learning efforts, in an integrated operations framework.

4.1 The rig control loop

In (Saputelli et al., 2002) the *Field Operations Hierarchy* was introduced (Figure 3). It gives an illustration of an oil field command chain, where information is relayed upward in the chain and orders are passed downwards, from the level of field life-cycle management to the real-time control of pumps.

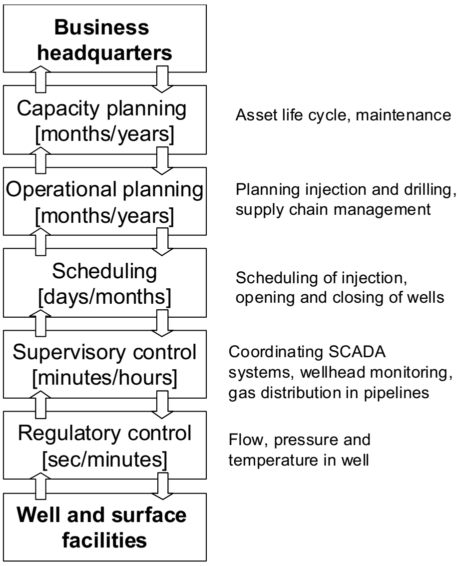

Figure 3: The Field Operations Hierarchy. Figure adapted from (Saputelli et al., 2002)

Two important observations can be made. First, the natural scales of time and space of the different levels increase upwards. To react to an imminent kick, one must take action within seconds or a few minutes, while the placement of the well itself may have been discussed for months before it is relayed down to the drilling team. Similarly, the kick is concerned with one well, while well placement may have taken the whole field into consideration. Sticking to this hierarchy simplifies the optimization of tasks on each level and makes them manageable both from a human and a computing perspective. The orders of the level above can be taken as fixed and it is assumed that optimizing each given task separately will bring about an optimized result on the level above.

This assumption does not always hold in practice. For instance, trying to maximize the production from each well may not necessarily maximize the production of the field as a whole. The practical solution to this is to develop simplified models of each well and their interactions These models can be used in planning and scheduling of production even if the well models would be too crude for real-time control of each well. This brings us to the second feature of the field operations hierarchy: The degree of abstraction changes between the levels. The lowest level deals only in numeric and binary data, while further up this is aggregated into more symbolic and abstract information such as "open valve", "pressure within safe limits", "gallons per minute", "daily production" and "net present value".

The natural separation by time scales is not only a feature of command chains but also of engineering issues and physical processes. Changes in pressure and fluid flow happen on a timescale of seconds while equipment wear happen on a timescale from days to months and issues of reservoir exploitation on a timescale of years. According to Saputelli (2002) the command chain is then actually induced from the different timescales of these processes.
Each level of the Field Operations Hierarchy will also differ in what models and machine learning methods are applicable. The hierarchy points out what the relevant time-scale is likely to be at that level, what simplifications may be warranted, whether numerical or symbolic data-analysis will be the most relevant and how fast our methods will need to be. The needs of the human operator also changes between levels, which must influence what output the AI produces.

The link between the levels and AI requirements are further illuminated by the theory of supervisory control systems which we now turn to.

4.2 Supervisory control systems

In recent years, the upstream oil industry has drawn inspiration from the oil refining industry and its integration of operations. Process units, plants and even entire supply chains are being monitored, controlled and optimized. A foundation for this integration has been to split up the optimization problem at different levels of detail and timescale. We exemplified this by the work of Saputelli (2002).
For this, the oil refinery industry makes use of supervisory control theory, which underlay much of the supervisory systems in various processing and manufacturing plants.

(Sheridan, 1987) identified five main tasks of the human operator in supervisory control. This is depicted in Figure 4.

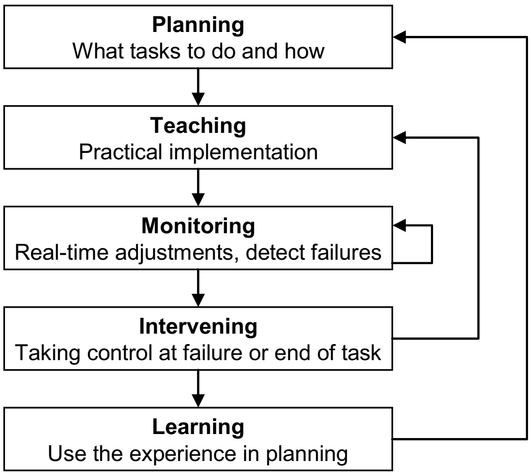

Figure 4: Human supervisory control. The system contains three loops with different time-scales. Figure adapted from (Sheridan, 1987)

Planning is the high/level scheduling of resources to achieve goals and to determine what is to be considered abnormal situations and plan ahead how to deal with them.
Teaching signifies the implementation of these plans.
Monitoring includes the real-time observation and minor adjustments needed to keep the plan on track.
Intervening include both taking over control when a task is at end and taking control when abnormal situations occur.
Finally, *learning* is using past experience to update plans and models. These tasks have three layers of feedback loops with increasing time scales from the innermost and outwards. These can be associated with the feedback loops in Figure 3 where data is transmitted to a higher layer and looped back as plans and orders.

Figure 4 could also be taken to depict the mental process of the individual human operator. This is discussed by (Rasmussen, 1986) which, as seen in Figure 5, define three types of behaviour:

Skill-based behaviour takes place after stating an intention and is done with little conscious deliberation. An example would be the skill needed to steer equipment using a joystick.

Rule-based behaviour is behaviour based upon certain process patterns and associated actions to be taken. These behaviours can be described verbally by the human operator but not necessarily explained.

Knowledge based behaviour has explicit goals and derived detailed process knowledge and reasoning around it. Plans and strategies are selected based on their effect on the stated goal.

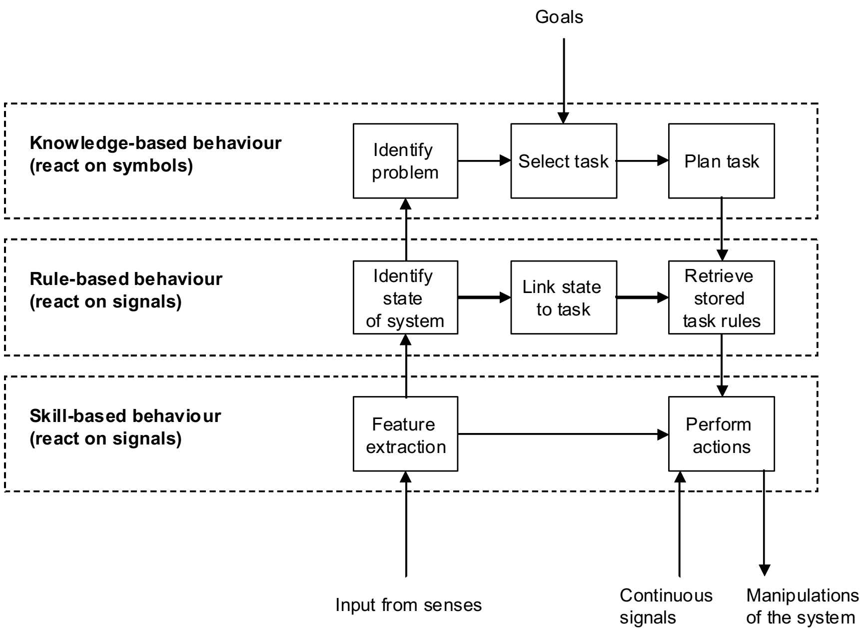

Figure 5: Characterisation of the operator's human behaviour. Increasing time scales upward. Figure adapted from (Rasmussen, 1986).

In the model of (Rasmussen, 1986), the translations from signals to symbols we saw in the Field Operations Hierarchy are made explicit.

4.3 Implications for our design

The theories and structures we have surveyed in this chapter carry some implications for the kick and stuck pipe alarm systems. The time-scale of a kick seems to confine its analysis to the lower levels of the hierarchy, implying that using purely numerical methods is the most suitable angle of attack. The output of the AI must be simple enough to fit the operator’s mode of work. The kick alarm helps to identify the state of the system and can be linked unambiguously to a task that the operator quickly retrieves and then performs.

The hierarchical models do not in themselves form a proof of this claim. But as outlined in Paper III, it is interesting to note that published work on machine learning methods in petroleum production by and large follow a pattern of numerical methods on the lower levels and symbolical ones such as case-based reasoning (Skalle and Aamodt, 2005) on the higher levels. This trend we believe, can be understood not only by the available data. But also by what responses the human in the control loop can perform on the different time-scales. Case-based reasoning in its original inception, presents the operator with a diagnosis based on causal chains. A form well suited for discussions and learning, but less suited for interventions in a manner of seconds. That numerical and symbolical methods are complementary is no surprise, but the theory of supervisory control systems seems to outline how such methods could most beneficially be integrated with each other. The greatest challenge for such an integration is probably to present suitably aggregated information to the higher level. Aggregation is a problem that can be framed as both feature construction from the time series and as classification and this is a topic we will be visiting later in this dissertation.

5 Fault detection and isolation

Fault detection and isolation (FDI) is the branch of control system engineering concerned with methods for monitoring a system through its sensor readings and accurately identify faults that occur. In this chapter we give a summary of the textbook-theory on observer based fault detection and survey recent attempts at extending this theory.

A typical approach to FDI is to set up a model of the system which predicts its outputs during normal operations. The analysis then centres on the *residual*, which is typically defined as the deviation between the predicted and real outputs.
An initial challenge is to separate deviations due to faults, from deviations due to measurement noise, benign variations in the system and deviations due to the model being an inaccurate description of the system. Fault isolation is concerned with categorizing the type of fault deviation, in other words to establish where in the system the fault has occurred. Formally, we may write a general nonlinear system as (Garcìa and Frank, 1997):

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t), \theta_f, \theta_d\right) \\ x(0) &= x_0 \\ y(t) &= h\left(x(t), u(t), \theta_{fs}\right) \end{aligned} \tag{5.1}$$

Where $x(t) \in R^n$ is the state of the system, $u(t) \in R^p$ is the input, $y(t) \in R^m$ the output and $\theta_f \in R^l$ the system parameters. We take faults to be unacceptable parameter values of the system, so that $\theta_f = \theta_{f0}$ when no fault is present. θ_{fs} represents parameters in the output equation, allowing for sensor faults which do not affect the system per se. Finally, θ_d represent mismatches between our model and the true system. Faults and model mismatches may also be defined as *unknown inputs* typically as extra terms in the equation.
We may then try to find a *residual generator* $r(t)$ of the form (Garcìa and Frank, 1997):

$$\begin{aligned} \dot{z}(t) &= g\left(z(t), y(t), u(t), \theta_{f0}\right) \\ z(0) &= z_0 \\ r(t) &= R\left(z(t), y(t), u(t), \theta_{fs0}\right) \end{aligned} \tag{5.2}$$

And a threshold:

$$th\left(y(t), u(t), \theta_{f0,} \theta_{fs0}\right) \tag{5.3}$$

Which satisfy the following inequalities:

$$r(t) \le th \Leftrightarrow \theta_f = \theta_{f0} \text{ and } \theta_{fs} = \theta_{fs0} \tag{5.4}$$

$$r(t) > th \Leftrightarrow \theta_f \ne \theta_{f0} \text{ and/or } \theta_{fs} \ne \theta_{fs0} \tag{5.5}$$

$$r_i(t) > th_i \Leftrightarrow \theta_{f_i} \ne \theta_{f0i} \text{ and/or } \theta_{fsi} \ne \theta_{fs0i} \tag{5.6}$$

Inequality (5.4) and (5.5) define fault *detection* by indicating that the value of r should be below the threshold if and only if no fault has occurred and above it if and only if at least one fault has occurred. Inequality (5.6) defines the fault *isolation* by demanding that an element of r is above the threshold if and only if a specific corresponding fault has occurred. The inequalities also capture the insensitivity to modelling errors θ_d.

5.1 State estimation

We may take $\dot{z}(t)$ to be a model of the system and h to be known from the model. We may then write a simple residual as:

$$r(t) = y(t) - h\big(z(t), u(t), \theta_{fs0}\big) \tag{5.7}$$

That is: r is equal to the measured output minus the predicted output from a simulation assuming no system or sensor faults. In this case the residual and the output estimation error, the discrepancy between measurement and prediction, is the same. The threshold th may be set as a constant. In chapter 7 we discuss offshore alarm systems that are based on this and similar residuals.
(Ding, 2008) observes that residual evaluation and threshold computation receives surprisingly little study, but list statistical testing and norm based methods as common approaches. An example of a typical norm based method is the evaluation of equation (5.4)-(5.6) with r defined as the root mean square of (5.7). The threshold value th may be the peak value of r under normal conditions. Replacing r with a windowed $\dot{r}$ allows for trend analysis (Ding, 2008).

Observers are defined (Westphal, 2001) as systems that try to approximate the state vector of some other system, using its inputs and outputs. In the original definition, the observer's state vector should be linearly related to the approximation, i.e. $z = T\hat{x}$. The term *identity observer* refers to those observers where $z = I\hat{x}$.

It has been shown (Frank and Ding, 1994) that all linear, time invariant residual generators which satisfy (5.4)-(5.5) can be brought into the form:

$$r(t) = R(t)(y - \hat{y}) \tag{5.8}$$

Where $\hat{y}$ is the estimate of y delivered by an identity observer and $R(t)$ is a parameterization matrix acting as a post-filter.

The problem of initial values and model uncertainty means that the observer can not be expected to make a perfect prediction of the internal states of the system from the inputs alone. To correct for this, the residual generator (5.2) allows output measurements to affect the state predictions. This is termed *output feedback* and can be seen in for instance the nonlinear identity observer (Frank, 1990) which is defined as follows:

$$\begin{aligned} \dot{z}(t) &= f\big(z(t), u(t), \theta_{f0}, 0\big) + L(z, u)[y - \hat{y}] \\ r(t) &= y - h\big(z(t), u(t), \theta_{fs0}\big) \end{aligned} \tag{5.9}$$

Here $\hat{y}$ denote output estimate. The matrix L, often referred to as the *observer gain* or the *Luenberger matrix*, should be chosen so that the state estimation error $e(t) = x(t) - z(t)$ is asymptotically stable at $e = 0$. That is, initial estimation errors die away. Other approaches, like nonlinear unknown input observer or disturbance

decoupling nonlinear observer are alternatives as long as f is assumed to be on specific forms (Garcia and Frank, 1997).
Observers employing the stabilizing term in (5.9) are called Luenberger observers. As pointed out by e.g. (Chaves and Sontag, 2002), Luenberger observers may be termed *deterministic Kalman filters* as they amount to Kalman filters which do not take noise statistics into account. Kalman filters in turn, are probably the most widely used state observers.

5.2 Redundancy

Instead of estimating the system state, one may attempt *parameter estimation*, where one tries to find a θ_f which results in a model matching the observed outputs. Faults are then classified based on the values of θ_f. For instance, one may try to estimate the friction parameter for a model of some rotating machinery. If the friction is high, this may indicate faults such as loss of lubricant or excessive wear.
Another method is the *parity space* approach. Given the equations for a model of the system, a vector called the parity check is constructed. This vector will be zero as long as the model describes the system faithfully. The parity space approach is in principle equivalent to the observer based approach (Magni and Mouyon, 1994) (Frank et al., 2000).

All these approaches may be said to employ what is termed *analytical redundancy*, to contrast it with *sensor redundancy*. If a sensor measure a given property of a system, adding an extra sensor to measure the same property would be redundant. However, if after a while the two sensors start reporting different values, this would be a clear signal of sensor fault. Similarly, known relations and correlations between different sensor readings give a form of redundancy. Imagine for instance a sealed container of a fixed volume filled with a known amount of high-pressure gas. This container is fitted with a temperature sensor and a pressure sensor. Given the pressure readings, the temperature sensor is redundant as the temperature could instead have been calculated using the ideal gas law $PV = nRT$. But given only the pressure readings, we might be unable to distinguish between a small leak in the container and a drop in temperature. With both the measured and predicted temperature available, a discrepancy between the two would signal a leak.
One may further distinguish *temporal redundancy* using redundancy between measurements from one or several sensors at different times.

5.3 Banks of observers

To facilitate a robust fault identification, it is common to use a bank of observers, which can be described schematically as in Figure 6. The system under surveillance has p inputs, *m* outputs and we have set up n observers. Observer #*i* may receive some system input and output as its input and at the same time makes a prediction $\hat{y}_i$ of the system output. Observer #*i* then produces a residual $r_i = \{r_{i1}, r_{i2}, \cdots, r_{im}\}$ from $\hat{y}_i$ and the system output $y = \{y_1, ..., y_m\}$.

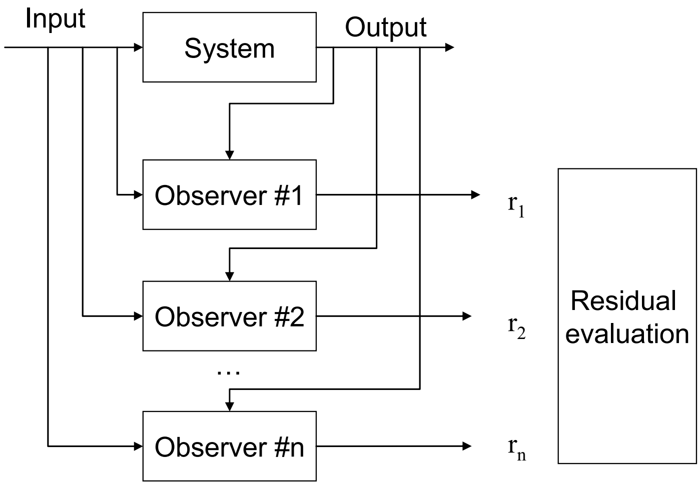

Figure 6: A bank of observers producing n residuals

Following (Frank, 1990) and using sensor faults as an example, we can distinguish between two types of observer banks:

In the *dedicated observer scheme*, we have as many observers as there are outputs, so $n = m$. Observer #*i* receives all system inputs but *only* system output y_i as its input. Following equation (5.9), the observer corrects its internal state using y_i. The observer tries to predict all or most outputs. If y_i is measured by a faulty sensor, the observer is misled about the internal state of the system. Analytical redundancy means that this will affect several of the observers' predictions and in turn the components of r_i. A nonzero residual r_i is then an indication of a fault in sensor #*i*.
In principle, this dedicated observer scheme allow us to detect faults in several sensors simultaneously.

In the *generalized observer scheme*, observer #*i* receives *all* system outputs *except* y_i as input. The residual is as before determined from a prediction of all or most system outputs. Again following equation (5.9), the system outputs are used to correct the internal state of the observer. The observer is therefore vulnerable to faults in all sensors except sensor #*i*, which it is *robust* to. When a fault in sensor #*i* occurs, observer #*i* is the only observer which still has a zero residual, (ignoring the component r_{ii}). This scheme, summarized in Table 1, only allow us to detect one sensor fault at a time. But the scheme is nonetheless more robust to unknown inputs affecting the true system state, since each observer is more thoroughly informed about the system.

	Observer 1	Observer 2	Observer 3	Observer 4
Fault 1	0	1	1	1
Fault 2	1	0	1	1
Fault 3	1	1	0	1
Fault 4	1	1	1	0

Table 1: A one in row j, column i, indicate both that observer #*i* receives input from sensor j and that its residual will be above the threshold during a fault in sensor j.
Sensor faults are uniquely identified by the specific pattern of residuals in each row, as long as no more than one sensor at a time is faulty. A fault in two sensors simultaneously would give the pattern 1 1 1 1 which do not distinguish the sensors.

Following (Alcorta Garcia and Frank, 1996), an observer is said to be *robust*[3] with respect to an unknown input d if the output of the observer is not affected by d.
If we want to include both y_i and r_i in the notion of "output", it is important to note that "unchanging" and "not affected" do not imply each other . As we saw above, the generalized observer #*i* produces a prediction $\hat{y}_i$ which is unaffected by a fault in sensor #*i*. But its residual component r_{ii} will be affected by the fault, since sensor #*i* enter in its calculation.
The observer may also be set to closely track the state measured by sensor #*k*, so that we always have $\hat{y}_{ik} \approx y_k$. In this case, the observer is not robust to faults in sensor #*k*, but r_{ik} is unchanged during a fault, since $r_{ik} = y_k - \hat{y}_{ik} \approx y_k - y_k = 0$.

It is interesting to note that in the case of a fault in sensor #*i*, the prediction $\hat{y}_{ii}$ by a generalized observer may serve as a substitute for the faulty sensor. This falls under the topic of virtual or soft sensors (Fortuna et al., 2007).

Output feedback is further discussed from a machine learning vantage point in (Yu et al., 1999) who call observers with and without output feedback *dependent* and *independent* models respectively. Dependent models tend to track faults, as was the case with r_{ik} above, but (Yu et al., 1999) also points out that they can only make one-step-ahead predictions since they require constant feedback from the real system. An observer with one-step-ahead prediction can in principle be made with no internal memory of previous states. Predictions with a trained feedforward neural network would be an example of this.

If we "short-circuit" the observer by feeding it its own predictions as input in the next time step, the model runs as an independent observer and can in principle make predictions arbitrarily far into the future. These observers possess a memory of past states by way of the feedback loop. The trouble with independent observers are that small errors in the model are allowed to accumulate in successive time-steps and the independent model runs further and further away from the true system state. Known as *drift*, this is a shared experience in all fields concerned with time series prediction, be it the study of statistical or machine learning prediction methods, industrial control or computationally intensive weather prediction

[3] Alcorta defines robustness for "a system" including the observer as a system in itself

5.4 FDI in Nonlinear systems

When the system evolution $\dot{x}(t)$ can be described by linear equations, some or all of the inequality demands in (5.4)-(5.6) can be satisfied using well-established approaches. When the system is highly nonlinear so that linearization of the problem become unfeasible, one needs to construct nonlinear observers, which has long been recognized as a challenging task (Bestle and Zeitz, 1983). As most systems actually tend to be nonlinear, the attention has in recent years shifted to FDI with nonlinear models. For a survey, see (Alcorta García and Frank, 1997). The surveyed methods dealt with specific classes of nonlinear equations, but did not address FDI for systems modelled by a general nonlinear equation. This might be characteristic of nonlinear problems, where solutions can not be combined via the superposition principle. Specific solutions is often desirable, but these FDI methods do not generalize to our case, where the e-drilling model is not only nonlinear but not even directly available as a set of differential equations. It might therefore best be treated as a black box.

An underlying question has been if it is possible to construct a method for residual generation and fault detection, which is valid for any nonlinear system. This had been stated as a long-term goal, but comparing (Alcorta García and Frank, 1997) and (Gayaka et al., 2007) one may sense a shift away from this aim, acknowledging that nonlinear systems are far too diverse to fit into a generalized framework.
Still, for “difficult” nonlinear systems, it is not unusual to see data-centric and AI-like methods being used (Frank et al., 2000). In many of these cases, a good model is simply not known. One solution could be to train a neural network to predict system behaviour using input and output recorded during the absence of faults. I.e, the AI takes the place of $g(\cdots)$ in equation (5.2). Focus may also be on the problem of distinguishing noise, modelling errors and harmless deviations from true faults, in which case the AI takes the place of R in equation (5.2) and the subsequent thresholding. These two uses may of course be combined, as seen for instance in the methods of (Terra and Tinos, 2001).

One use of historical data may be illustrated by the sealed container example above. Deviations from the ideal gas law could create a mismatch between measured and predicted temperature even in the absence of a leak, so that model mismatch and true faults could not be disentangled. These deviations would however occur repeatedly in recorded time series and could be learned by a variety of AI methods.
An example of handling model mismatch can be found in Paper I where we used the residual in equation (5.7) and incorporated learned model-mismatch into the threshold function (5.3).

6 Problem characteristics and the choice of method

Machine learning offers a bewildering range of methods and tools for data analysis. Many of these are implemented in freely available software packages, but it may be necessary to build up a familiarity with the algorithm to tailor the pre-processing, successfully tune the parameters and interpret the results. Regarding the choice of methods, we discussed earlier in this dissertation that for real-time drilling problems, we should focus on methods dealing with numeric data, at the expense of more symbolically oriented methods.

Even under this condition, it is within our means to test a large number of methods each with a wide range of settings, collectively referred to as AIs. Using the available time series, we could simply select the one with the best performance. This would however be problematic in the light of chapter 3.1. With a large number of methods, it is likely that some will perform very well on the test set by chance, a well-known problem in data-mining. As we test more methods, the performance of the best AI on the test set will be a more and more inflated measure of its true performance (Hand et al., 2001). This remains a problem with cross-validation. We may use an independent validation or hold-out set for true performance measurements. If this set reveal that we have chosen a bad AI, we would have to pick new AIs to test. But picking from a collection is what inflated the performance measure in the first place, this ruins the use of the hold-out set.

Large test and validation sets also eats into the size of the training set. Thus instead of testing many methods, we take this fork in the path as an opportunity to introduce a-priori knowledge. In this chapter we discuss ways in which this can be accomplished.

6.1 Limits brought on by the raw data

The historical data available to us consists of daily reports by the drilling crew and a multivariate time series which has typically been sampled once every 1 or 5 seconds. Important measurements include pressure in the well, speed of rotation of the drill string, pump rate, depth of the well and torque on the drill string. A typical drilling operation may stretch over days or weeks, resulting in a correspondingly large data set. A typical representation of a drilling time series is seen in Figure 7. The daily reports provide a rough annotation for this time series, indicating when a drilling problem occurred and providing a summary of larger time periods. There is however no sample-by-sample annotation of the time series. The different drilling operations or *drilling modes* are only specified by approximate times in the log, while routine operations such as making a connection, is omitted. The time series are therefore mostly *unlabeled* examples. It is possible for an expert to label some of the modes manually and some modes can be identified automatically, at least in retrospect (Thonhauser, 2004, Thonhauser and Mathis, 2006). But labeling of faults is more difficult. Some faults have an unknown origin, were misinterpreted, or evaded detection altogether by the drilling crew.

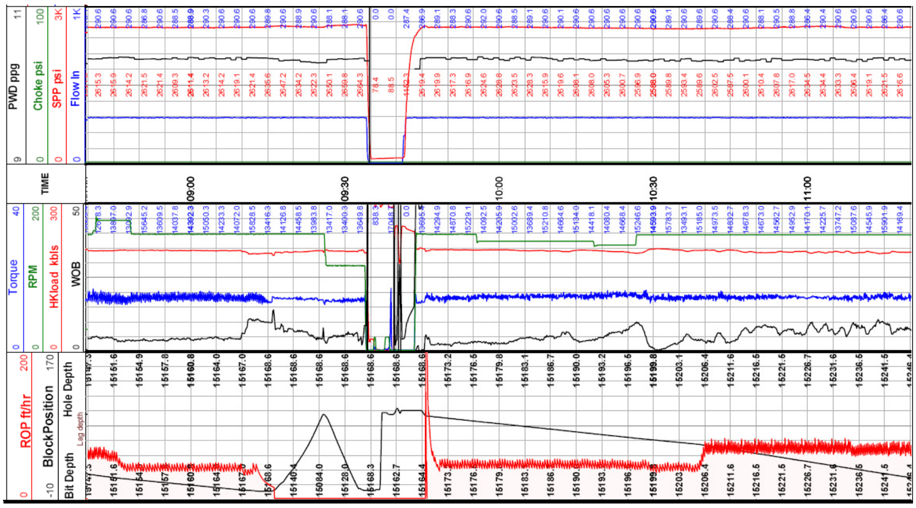

Figure 7: Typical presentation of the multivariate time series resulting from a drilling operation. An erratic torque (centre blue) indicates poor hole cleaning. The hole is subsequently cleaned with bit off-bottom.

As for other characteristics, drilling experts are likely to list noise as the number one characteristic of drilling time series. Calibration errors and inaccurate measuring devices are other common issues. Also, the low sample rate means loss of information in the high-frequency range. Intermittent loss of real-time downhole signals was until recently an unavoidable part of real-time operations, as data-transmission from downhole instruments were performed using pressure pulsing driven by the mud flow. The transmission then ceased for as long as the pumps were turned off, though continuous recordings may be retrieved from the instrument later on. This is set to change with new technologies like wired drill pipes, giving a continuous and much larger downhole bandwidth.

The characteristics mentioned above imply that supervised learning of the time series need to be tolerant to noise and incorrect labeling or be able to learn from only a small set of trustworthy cases.
In a typical drilling operation, the drilling mode will shift many times and characteristics such as pressure and rate of penetration will change as the well is expanded. The number of drilling problems will be few compared to the number of these harmless and normal events, which creates a strong *class imbalance* (Chawla et al., 2004), an obstacle for some computational learning methods. Counted by events and not by samples, we are actually limited to a rather small training set, which in turn, according to VC theory, (see chapter 3) limit the allowed complexity of the AIs.

The severity of this can be quantified. Time series prediction usually involves a sliding window approach where the d latest samples are used to predict the value or state a few steps ahead. The number of samples in the window then determines the dimensionality of each example. Taking a neural network as an example, a fully connected feedforward neural network with a d-dimensional input, h nodes in the hidden layer and one output node, has $d*h+h$ free parameters, which roughly equal its VC dimension (Theodoridis and Koutroumbas, 2006) . With n the number of examples, we must then require that:

$$n \gg d * h + h \tag{6.1}$$

A best case scenario would be one with 50 examples, half of which are drilling faults and half no fault and the faults are under similar conditions. We may use a window of 12 samples, which is equal to one minute if we are sampling at 1/5 Hz. It then follows from equation (6.1) that h can be at most 2 or 3, a severe restriction on the complexity of the network. Meanwhile, equation (3.1) indicates $\varepsilon \approx 0.72$. This means that if the 50 examples were sampled evenly in a 12-dimensional hypercube of length 1, the distance between neighbours could be as high as 0.72. This goes to show the non-intuitive nature of high-dimensional spaces and implies that the accuracy would be disastrous for most function approximations.

6.2 Remedies

The most obvious remedy to the above problem is to stick to simple methods and do some form of dimensionality reduction on the sliding window, but it is also possible to multiply certain examples. In Figure 7 we see poor hole cleaning with a recognizable signature over half an hour. As poor hole cleaning is a precursor to a class of stuck pipe incidents, we may count near-faults among the examples. Further, the half-hour signature can be split into several shorter examples, albeit not wholly independent ones. In Paper IV we illustrate the above remedies, with further discussions in chapter 10.2.

We may also cast the problem in a new mould. Rejecting events as the basis and instead focusing on the individual sample will increase the number of examples available. It is possible to train an AI to predict one variable such as pressure based on other variables and a deviation from the prediction could serve as an anomaly detection (Fruhwirth et al., 2006). This was in essence our approach in Paper I and Paper II where the AI needed no labeling of the raw data.

The time series available to us may be from one or several wells. But if an AI is used for monitoring a drilling operation, the well being drilled will naturally not have been in its training set. One should therefore use a new well as the test or validation set, to get a more reliable assessment of the AI's field performance. The need to prepare the AI for an unknown well has some additional implications which we will discuss in chapter 6.4.

6.3 Interesting and uninteresting correlations

Among the correlations we may find with faults, we make a distinction between what signals an imminent fault and what causes it. For instance, a certain choice of drilling mud and a low pump rate may lead to inefficient removal of cuttings and therefore an increased chance of stuck pipe. This is important and could possibly turn up as a correlation in the time series. But this is also information that would be available at the planning stage as soon as mud type and pump rate had been decided on. An AI based on this correlation would not be able to distinguish between bad and sufficient hole cleaning in a real-time situation. In other words, this correlation only gives an a-priori probability. Torque on the other hand, is *affected by* poor hole cleaning. Thus we are able to assess the likelihood of a fault, a posteriori the measurements.
As our task is detection through real-time analysis, we try to avoid picking up a priori relations, deferring these to planning-stage data mining. Some variables may of course feature in both stages, such as e.g. a pressure drop, which may both indicate a kick and be the cause of it.

6.4 Model selection and the true model

Once a model has been chosen, the estimation of its parameters and a measure of their uncertainty can proceed by well-established methods such as maximum likelihood, least squares and gradient descent. This is in essence an optimization problem. The choice that has to be made between different models precedes the optimization process and has until now received less attention. As we described in the previous chapter, the shotgun approach of trying out all models has become feasible with modern computers, but the method is unreliable and rules of thumb like parsimony and incorporation of prior knowledge is necessary.

Model selection is discussed in depth by (Burnham and Anderson, 1998). They focus on biology, but their examples also bring out the challenges in our data gathering and model selection in a surprisingly clear manner. We therefore recount parts of their argument here.

If one wishes to study the properties of a mixture of two chemicals, a sound plan is to observe it at different combinations of pressure, temperature and ratios. We make sure that the experiments evenly cover the parameter space of the model. This was also an assumption in equation (3.1). A biologist may want to produce a model of population dynamics and make observations by counting animals in an area, recording migratory patterns and observing variables thought to influence this, such as temperature, availability of food and the number of predators. These are conditions that can not be reproduced experimentally and some combinations of parameters may never occur. The observations may in the worst case cluster in a corner of parameter space. This inability to make controlled experiments is a defining characteristic of an *observational science*.
Our time series analysis of drilling operations and indeed much of data mining, fall into this category. While the properties of the drilling fluid is measured experimentally for much of its relevant parameter space, we are in no position to e.g. provoke a stuck pipe incident by varying drilling fluid viscosity, rock properties and pump rate. The behaviour of our models and AIs in badly surveyed parts of the parameter space therefore becomes of great concern.

In experimental sciences an experiment might be repeated under different conditions while holding the relevant parameters unchanged. If the outcome changes, it could indicate that a new uncontrolled parameter is having an effect. In an observational science we have no guarantee of repeated experiments and it is more difficult to know if our model contains all the relevant parameters. That is, the dimensionality of the true input space is not necessarily known.

Suppose for a moment that the dimensionality is known and that we have enough evenly spaced observations. One would then think that it is possible to approximate the true relationships between the variables. This rests on the underlying assumptions that "a true model" exists and that our models, be it AIs or parameterized physical models, converge towards it. These assumptions might be reasonable in cases where the true relationships are known to be relatively simple. Methods such as Maxium Likelihood parameter estimation do converge asymptotically to the true values, under certain conditions (Theodoridis and Koutroumbas, 2006).
That these assumptions hold in general is something (Burnham and Anderson, 1998) among others, take issue with. Namely, the true model may not exist. If it exists it may not stay fixed and there may be a very large number of parameters to determine. This is illuminated by (Burnham and Anderson, 1998) with an example from population dynamics:
We assume that there exist 10 years of observations of a specific colony of seagulls on a given island. We want to model the population dynamics but would like more data to test and fine-tune our model. We may then import data from colonies on other islands or other species of birds. But by doing that, species and location enter as variables that must be accounted for. The underlying model thus gets more complex the more data we add.
Our task of learning from drilling time series faces a similar problem. In preparing for a new well, we may first train our system on recently drilled wells from the same field. To increase the number of samples we also include wells from fields with a different geology, drilled using a different type of mud etc. The dimensionality of the problem is therefore increasing, unless we know we can discount them or use a greybox approach and assume that the new parameters are fully accounted for in the physical model. This is unlikely to be the case in stuck pipe incidents, where the interplay between rock type and mud chemistry is still an active research topic. (Burnham and Anderson, 1998) argue that the dimensionality of the true model may well be infinite, only restricted when we discount smaller deviations or limit our observations to fewer environments. Oil exploration is a lot simpler than biology, but even here we can at least make the case that the dimensionality is not bounded. The rapid technological developments in drilling equipment, new control strategies and exploitation of successively deeper wells with higher pressure and temperature means that new wells will enter territory not seen in the old training set and be described by parameters not yet defined.
A field deployment of an AI-based alarm system will have to take this into consideration. One will have to rely on drilling simulators to test its performance in novel conditions and the training set may be further tailored to the task.

7 Alarm systems in drilling

We will now give a short overview of the approaches taken to alarm systems used in the drilling industry. In (Hargreaves et al., 2001) the tools for supervision and diagnosis during drilling were laid out as developing over three generations:

1st generation: Pre-determined threshold values define normal operation. For instance: If the pit volume exceeds a given level a kick may be in progress.

2nd generation: Noise tolerance is taken into account and more than the very last measurement is used to define the threshold. I.e. the threshold is windowed. The CUSUM method (Basseville and Benveniste, 1986) is perhaps the best example. This method detects step changes in a variable but is not well suited to detecting gradual changes, e.g. the typical ramp shape of a kick. It also relies on the common assumption that the noise is constant-variance Gaussian, which may not be the case.

3rd generation: The third generation introduces predictive modelling. The false alarm rate is reduced because safe predictable events can be ignored. However, trend detection is still done using a windowed threshold. This does not take into account the shape of the trend nor changes in noise. Neither are rig-dependent effects taken into account or they require accurate calibration. *Heteroscedasticity* or non-constant variance, remains a problem. Heteroscedasticity in turn, is one of the ways in which the stationarity assumption is violated. (Palit and Popović, 2005).

These alarm systems can be understood within the framework of chapter 5 as systems that analyse a discrepancy between a predicted and a measured value. The first generation system implicitly predicts no change in pit volume and raises an alarm when measurements deviate from this beyond a threshold value. We may write this as

$$\begin{aligned} \dot{z}(t) &= 0 \\ z(0) &= z_0 \\ r(t) &= y(t) - z(t) \end{aligned} \tag{7.1}$$

With a constant threshold

$$th(t) = c$$

A second generation system using CUSUM can be written on the form

$$\begin{aligned} \dot{z}(t) &= 0 \\ z(0) &= z_0 \\ r(t) &= R\left(r(t-1), y(t) - z(t)\right) \\ \\ th(t) &= c \end{aligned} \tag{7.2}$$

For CUSUM, a process showing fluctuations around a mean is taken as the zero-hypothesis and an alarm is triggered when, by hypothesis testing, the mean is found to have shifted. In this way, stretches of the time series can be classified as containing or not containing a step change. (Refer to Figure 8.)

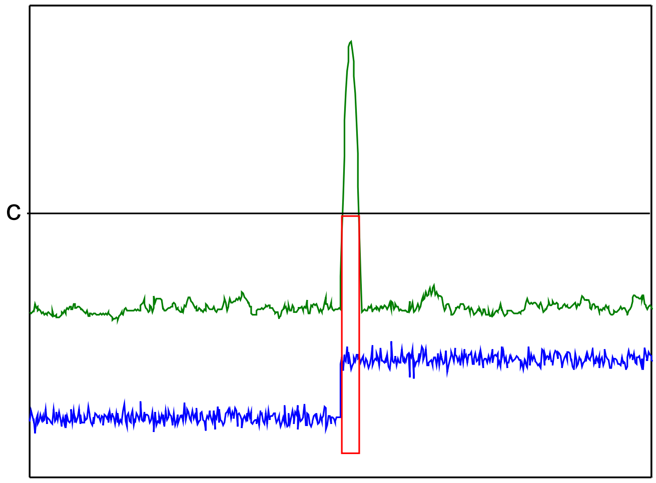

Figure 8: A process (blue) fluctuates around a mean. A simple windowed cumulative sum (green) detects a step change in the process. Given a threshold $th = c$, the time sequence within the red rectangle can be classified as a changepoint.

From Hargreaves survey, it appears that it is only in the third generation that models are stated explicitly and separated from the analysis of the residual. The third generation then conforms to the long-established (Frank, 1990) division between residual generation and evaluation. One where the physical model is restricted to the generating stage.
Meanwhile, the scope and sophistication of physical models in the petroleum industry have advanced steadily over the years. Physical models feature in the planning and optimization of wells, training of personnel and real-time control of oil field operations.
This is a tremendous advantage for third-generation alarm systems. The physical models can be plugged into the residual generation stage, models which have a level of sophistication and independent verification which would not have been feasible in a stand-alone alarm system.

The residual evaluation stage has in contrast not seen a comparable development in offshore alarm systems, with the related field of predictive maintenance as a possible exception. In the next chapter we will discuss if the problem could have been split up in a manner different from the one we have seen here.

8 A combined approach

Taking a machine learning view on the split between residual generation and evaluation, we realize that the generative stage is a prediction task while the evaluation stage is framed as classification. The latter is also observed by e.g. (Marcu et al., 2003).
It is interesting to note that the prediction/classification split runs parallel to another split, that between physical models and data-centric methods. Physical models tend to be employed for prediction tasks in the residual generation, while the data-centric methods are mostly confined to the residual evaluation stage.

We ask if this is a necessary design restriction, or if alternatives exist. An example of physical or *model-centric* classification would be to use expert knowledge to infer a fault. A framework for doing this on a large scale would be expert systems, but as we argued in chapter 4.1 expert systems are best suited at higher levels in the field operations hierarchy.
A data-centric prediction would seem attractive, with the possibility of learning from recordings of past drilling operations, but it would be a daunting task to create something matching today's physical models in accuracy. This is perhaps impossible with a purely data-driven method, given the state of the art of machine learning and the available data sets.
This does not exclude the possibility that a *combination* of physical models and data-centric prediction could together achieve better accuracy at the prediction stage. The simple greybox method explained below goes a long way in achieving this.

8.1 Greybox prediction

We recall from chapter 5 the residual defined as $r(t) = y(t) - h(...)$ where y is the system output and h is our model prediction. Rearranging the equation into $y(t) = h(...) + r(t)$ we see that we have effectively split the observed behaviour into an explained and an unexplained term. The greybox approach found in for instance (Forssell and Lindskog, 1997) proceeds by delegating the unexplained data to an AI and sum its result with the physical model. I.e. for a given training set we compute the residual:

$$r(...) = y(t) - h(u_t, ...) \tag{8.1}$$

This gives us a time series of pairs of inputs and residuals $\{u_t, r_t\}$ which is used to train an AI with these as inputs and outputs respectively. The combined prediction becomes:

$$\hat{y}(...) = h(u_t, ...) + \hat{r}(u_t) \tag{8.2}$$

Where $\hat{r}$ is the AI prediction of r. We may then form a new residual:

$$r_{new}(...) = y(t) - \hat{y}(...) \tag{8.3}$$

This residual r_{new} is what is left unexplained after an analysis by both the physical model and the AI. The procedure is described schematically in Figure 9.

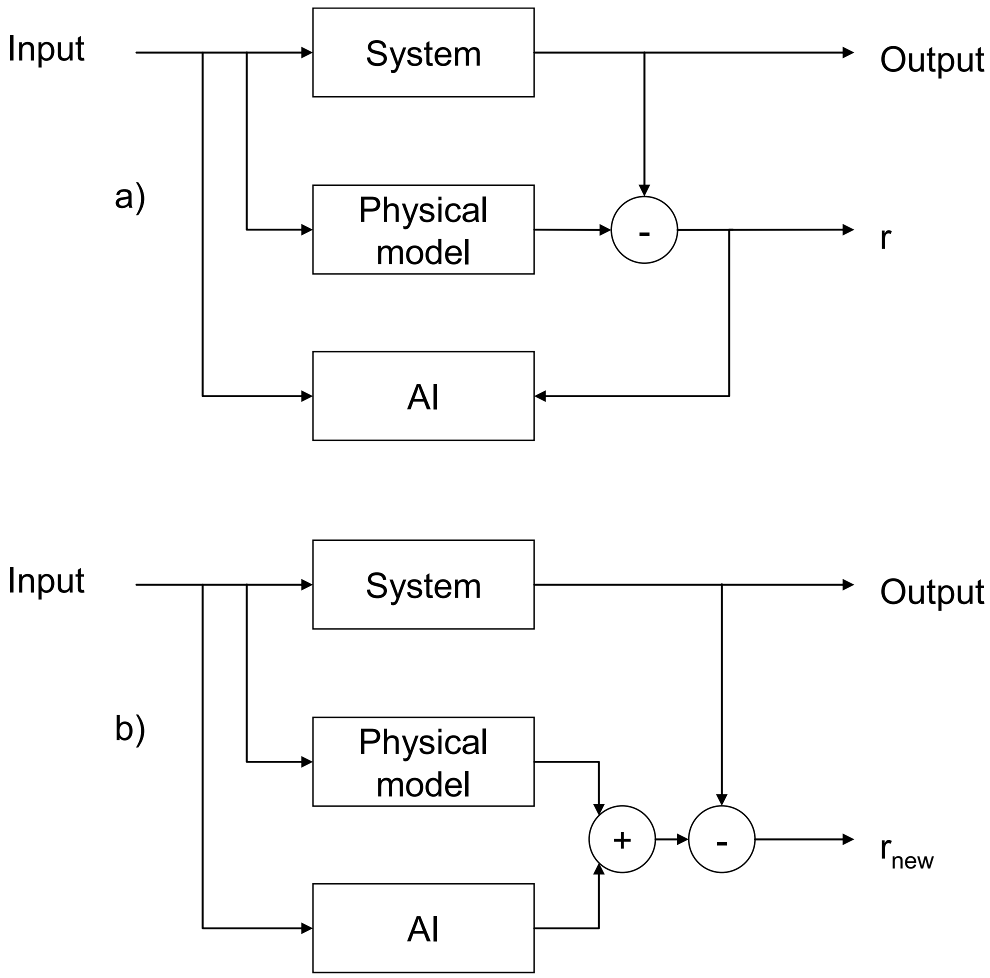

Figure 9: The greybox method with the training stage of the AI a) and the prediction stage b). In the learning stage the AI learns the errors made by the physical model. In the prediction stage, the output of the AI and the physical model is combined, yielding a more accurate prediction and consequently a smaller residual.

In Paper I we tested this approach on predictions of return flow rate, an important indicator of kicks. We found that given the residual and measurements of mud density and pump rate, the AI was able to learn a common harmless phenomenon that was unaccounted for in the physical model. Using the above approach the overall prediction was improved, reducing the number of false alarms generated in the residual evaluation stage. (Refer to chapter 10) A similar approach can be found in e.g. (Zak et al., 2001).

8.2 Quality criteria and greybox design strategies

During training, we need to decide how good the AI must be to improve on the overall prediction. The prediction is improved if less of the data remains unexplained, that is:

$$|r_{new}| < |r| \tag{8.4}$$

Rearranging equation (8.1)-(8.3), we find that $r_{new} = r - \hat{r}$ and it follows that we achieve an improvement if:

$$\hat{r} \in (0, 2r) \tag{8.5}$$

This would appear as a fairly lax bound, but may nonetheless be a challenge if r has sign-changes in the time series or if it varies widely in size while the AI has a noise term that is not a percentage of the output.

There is also an underlying assumption that it is easier for the AI to learn the residual than the full time-series, i.e. that the residual is “simpler”. This requires a good model h. With reference to equation (8.1), we want many of the terms in y to cancel or almost cancel with corresponding terms in h, so that the residual may be approximated by the remaining fewer terms. For instance, the term describing the well understood mud displacement by the drill-string will cancel, while the term in y representing draining pipes remain.

It is easy to construct examples where h is good, but where equation (8.1) does not produce a simplification. For instance if $y = c_y f(...)$ and $h = c_h f(...)$ with c a constant to be calibrated, the residual becomes $r = (c_y - c_h) f(...)$ which has all the complexity of the original time series. The obvious solution is to redefine the residual as:

$$r(...) = y(t) / h(...) \tag{8.6}$$

This is in line with the findings by (Forssell and Lindskog, 1997). They found that equation (8.6) could in some cases produce better results than equation (8.1). In a few cases it might be possible to guess the form y and r takes, typically where the residual is only due to parameter uncertainty in the model. In such a case, it might be possible to craft r to be as simple as possible, facilitating quick training of the AI. This translates into a r_{new} which is robust to parameter uncertainty, a criteria for residuals in FDI (see equations (5.4) - (5.6)).

Having produced r_{new} we transmit it to the residual evaluation stage. Here we again face the criterion that r_{new} should be better than r, for the purpose of fault classification. This demand may in fact sometimes conflict with the prediction criterion in equation (8.4). A hypothetical AI may produce a good prediction in the squared error sense, but add noise in a high-frequency range crucial to detecting a certain drilling problem. Nor is squared error alone a good check on the requirement that r_{new} deviates markedly during faults.

The choice of r is further complicated by the fact that terms may almost cancel only for certain states of the system. In short, what constitutes the ideal residual depends on the real system, the physical model, the choice of AI, and the specific drilling problem.

To have a chance at working with these conflicting demands, it would be preferable to employ different AIs simultaneously. In chapter 9 we return with a proposed architecture that incorporate this specification.

8.3 Restrictions and opportunities in residual design

We have now covered the requirements on the *output* of the residual function, namely simplicity and adhering to the requirements in FDI. For nonlinear observers, FDI do not specify further restrictions on the residual function itself. However, the greybox approach introduces an additional restriction. In equation (8.2) we are reconstructing the system data from the residual, the equation is effectively inverting the result of (8.1). So the residual r must be invertible.
That is, we require that:

$$\left[r,h\right]=f\left(y,h\right) \tag{8.7}$$

has the inverse

$$\left[y,h\right]=f^{-1}\left(r,h\right) \tag{8.8}$$

An invertible function has a one-to-one correspondence between the input and output and must be strictly increasing or decreasing. We see that equation (8.6) only has a partial inverse, for positive values of h. With f a general nonlinear function, a sufficient condition for invertibility is that the Jacobian of f is nonsingular (Renardy and Rogers, 2004). I.e. the Jacobian determinant should be nonzero.

It is not our intention to produce an overly complicated residual function, in fact we will mostly use equation (8.1) or other simple variations over equation (5.8). Instead we want to point out that many well-known operations that we think of as pre-processing of the data, form valid residuals. Scaling and shifting of the variables as well as rotation of the input space are all invertible operations and could be part of a residual. Principal Component Analysis (PCA) examplifies this (Nortvedt et al., 1996). PCA basically performs scaling and rotation operations in the input space to capture as much of the sample variance as possible with a projection onto two or three orthogonal axes. The residual variance remains as what is unexplained by these projections. Even some sliding window methods and filters can be implemented as invertible operations. An example is the invertible multivariate MIMO FIR filter described by (Rajagopal and Potter, 2003). This opens up for intuition developed in data pre-processing to be used in nonlinear observer design.

9 Proposed architectures

As explained in chapter 5.4, traditional ways to construct observer-based alarm systems are somewhat obstructed by insisting on a black-box model. Yet an architecture like the generalized observer scheme (GOS) is an appealing way to structure our a-priori knowledge about the system inputs and outputs and properties of the faults.
A simple solution is to replace the traditional observer with an AI. (Erdogmus et al., 2002) even shows equivalence between a certain AI and a Luenberger observer. From this it is in principle easy to construct a GOS for sensor or actuator fault detection. See for instance (Marcu et al., 2003) for an example of a GOS which employs neural networks in both observers and in the residual evaluation stage.

For sensor and actuator faults, the signatures in a GOS system are straightforward. But the isolation of a system fault requires a bit more system knowledge. A fault may or may not affect the output of more than one sensor, which must be taken into account when making observers robust to the fault. A generic approach could be to set up observers with all possible combinations of inputs and feedbacks. A sufficiently advanced pattern recognition working at the residual evaluation stage could then in theory detect and isolate faults by learning on this large set of residuals. Though feasible, it might be too complex to be reliably trained by the available data. Instead of packing the complexity into the AI algorithm, we will in this chapter attempt to take more of the classification complexity out into the GOS framework.

9.1 Training strategies for the AIs

In the traditional GOS, the role of each observer is defined by its set of inputs, output feedbacks, and which outputs it predict. When the observer is a black-box AI, we can sometimes also influence the feedback strength, but otherwise our involvement in the architectural design stops at this point.
Even though we have discussed the choice of AIs for drilling problems at length, it seems difficult to prescribe different optimal parameters for each different observer in the GOS. We therefore leave these considerations inside the black boxes.

The choice of training set for the AI on the other hand, appears as a highly relevant choice. A properly configured AI should be able to reproduce system behaviour if the system behaves similarly to how it behaves in the training set. As an observer, the AI will then produce a small residual. Conversely it *may* produce a large residual if the system is in a mode not seen in the training set. We may interprete this as the residual acting as a *similarity measure* between the stretch of time in the training set and the current state of the system. If we train several AIs on different training sets, we can take the resulting residuals and their thresholds as a *classification system*, with labels corresponding to the training sets.

The idea of cross-prediction error between data sets as a similarity measure, were to the author's knowledge first presented by (Hernandez et al., 1995, Schreiber, 1997). Schreiber (1997) uses the cross-prediction error for analyzing nonstationarity and parameter drift and in (Schreiber and Schmitz, 1997) also applies it to clustering of

the data sets. Clustering in itself is also well-known as an anomaly detector and has been studied for fault detection (Tanaka et al., 1995).
Our main motivation for introducing cross-prediction is that it acts as an *unsupervised* classification of the time series. As was explained in chapter 6.1, our time series are mostly unlabeled. Since trained AIs capture the dynamics of the system, it is quite possible that these classifications will be relevant and can be further interpreted in light of the input subspace of the AI.

In the GOS, this similarity measure could potentially add a richer structure. If the observer is trained on a time series containing only drilling mode A and the system progresses from drilling mode A to B, the observer would still predict A, resulting in an increase in the residual. We could then claim that the observer is *robust to the mode change*, taking the cause of the mode change as an unknown input.

We recall that training otherwise identical observers on different subsets of the available training set is how ensemble diversity is usually induced in ensemble learning. An alternative was the subspace method, where the AIs were trained with different subsets of the available inputs. The subspace method was originally used by (Ho, 1998) for a classification task and it is interesting to note the similarity with the classical GOS where the observers also work on input subspaces and are combined for a classification task.
We will not go so far as to claim that a GOS *is* an ensemble system, but there are similarities in their tactics and it appears relatively straightforward to integrate them. In ensemble terms, we claim that the GOS sketched above is enhanced in that it has *two* sources of ensemble diversity instead of only one.

We have not touched on how the residual evaluation stage should be carried out in the proposed architecture. This is a separate issue, but in the next section we will discuss whether or not ensemble methods could play a part in the evaluation stage as well.

9.2 Structured residuals for classification

We said that the residuals generated in our proposed architecture could be interpreted as similarity measures which in turn could be interpreted as classifications. The task of the residual evaluation stage would then be to combine and interpret these classifications into fault identifications. This fall under the topic of classifier combinations, which include some ensemble methods. (Ho, 2000) identified two parallel lines of study in classifier combinations:

1. *Decision optimization*: Assume that we have a fixed set of specialized and carefully designed classifiers. Our task is then to find the best combinations of their decisions
2. *Coverage optimization*: Assume that the way we combine the classifiers is fixed. Our task is then to generate a set of mutually complementary generic classifiers that achieve an optimal accuracy.

Assuming that the residual generation stage is fixed, a subsequent discussion would lock us on to decision optimization. In this section we will not attempt to present an optimal classifier combination nor to give a full survey of the many methods for this.

We only seek to demonstrate that the residual evaluation part of the classical GOS yields as easily to an ensemble method analysis as did the residual generation part.

We start by recalling the fault classification in a standard generalized observer. As seen in Table 1 on page 23 and repeated below, each fault has a specific pattern of residuals:

	Observer 1	Observer 2	Observer 3	Observer 4
Fault 1	0	1	1	1
Fault 2	1	0	1	1
Fault 3	1	1	0	1
Fault 4	1	1	1	0

Table 2: Reprodution of Table 1.

Other sets of structured residuals are possible (Magni and Mouyon, 1994). We will here frame the problem in its equivalent ensemble form:
For each observer the residual evaluation issues alarm/no alarm by evaluating some threshold expression such as $|r_i| > th()$ which is either true or false. For each observer we then have a binary classifier. The generalized observer uses the table above to turn four binary classifiers into one multi-class classifier that can distinguish between four fault classes. While being a part of basic observer system theory, this form of classifier combination is also a very attractive method from a machine learning point of view, as setting up and training a binary classifier is often much easier than creating a multi-class classifier.

The classical GOS decision table has some well-known shortcomings. As we have mentioned earlier, Table 1 with its corresponding output feedbacks implies that the GOS does not handle two simultaneous faults, nor a misclassification by any of the observers. It is suitable to analyze this problem using the Hamming distance between the rows, as has been recognized both by the fault detection and isolation community (Staroswiecki and Comtet-Varga, 2001) and the ensemble community where (Dietterich and Bakiri, 1995) employed error correcting codes from signal theory. The scheme of the latter is called Error Correcting Output Codes (ECOC) and we will take a moment to explain their procedure:

9.2.1 Error Correcting Output Codes (ECOC)

We start by augmenting the original GOS table with three new classifiers:

	O1	O2	O3	O4	**O5**	**O6**	**O7**
Fault 1	0	1	1	1	1	0	0
Fault 2	1	0	1	1	0	1	0
Fault 3	1	1	0	1	0	0	1
Fault 4	1	1	1	0	0	0	0

Table 3: Table 1 with three new classifiers.

Each of the four rows in Table 3 now correspond to a code word generated by the Hamming(7,4) code. These have the property that a single mistake by one of the classifiers (a bit flip) can not turn one row into another.

For instance, a mistake in observer two during fault one results in [0 0 1 1 1 0 0] which is still closest to Fault1 measured by Hamming distance. Hamming(7,4) also guarantees that if two observers make a mistake simultaneously, this will be detected as an error.

The extra observers O5-O7 turn out to be dedicated observers receiving input from only one output sensor. This may not be desirable for some observers but we are free to choose a different set of code words, such as Table 4, where all observers get at least two output feedbacks.

	O1	O2	O3	O4	O5	O6	O7
Fault 1	0	1	1	0	0	1	1
Fault 2	1	0	0	1	0	1	1
Fault 3	0	0	1	1	1	1	0
Fault 4	1	1	1	1	1	1	1

Table 4: Alternative with more output feedback.

Note that Table 4 still retain the link between “Fault i” and “sensor i”. For instance, even though no observer is robust to a fault in sensor 4, it can still be uniquely identified from the table.

While Table 4 would seem to offer a straightforward improvement over Table 1, our treatment of observers as binary classifiers brings in its own set of issues.
The response of O6 in Table 4 may represent a plausible observer, for instance the dedicated fault detector in (Zhang et al., 2008). But as a classifier it is meaningless, unless we add the fault-free state as a fifth row in the table. Dietterich and Bakiri (1995) also recommend that the observers are uncorrelated, as a correlation could produce several simultaneous bit flips. They also specify that Hamming distance between the columns should be maximised. The rationale for the latter is that classifiers trained on similar classification tasks will themselves be similar and thus correlated. Similarity includes complements because binary classes are treated symmetrically by most algorithms. Observer 3 and 7 in Table 3 for instance, produce opposite classifications but have identical class boundaries.

It is not immediately obvious how correlation between observers and column distance would play out in our case. Ensemble diversity by training subsets go some way towards alleviating correlation, but it is also dependent on the hard-coded threshold functions. Worse is that ECOC has not solved the problem of simultaneous faults. If for instance fault 3 and 4 in Table 3 occurred simultaneously, this would be misclassified as fault 3 plus a bit flip.

To address these problems, we would have to construct longer code words and thus use more observers. The next section exemplifies an easy way to increase the number of observers with meaningful information, especially for distinguishing system and sensor faults.

9.3 The model as an observer

The pipe draining effect described earlier, is typically visible via only one sensor. It would therefore be difficult for a GOS to distinguish it from a sensor fault. We address this problem by introducing new observers. We first add the physical model as observer O1. It has model prediction of flow out as its prediction, thus its residual is predicted minus measured flow. We also add the system described in Paper I as observer O2. It receives two inputs: Pump rate and model prediction of flow out. It in turn produces a prediction of measured flow out, with a corresponding residual. The result is presented in Table 5:

	O1	O2 (AI)
Static pump rate	0	0
Kick[4] or AI fault	0	1
Pipe draining/filling	1	0
Kick or sensor fault	1	1

Table 5: O1: Observer similar to physical model, O2: AI pipe draining observer.

Combining observer O1 and O2 allows us to tell pipe draining apart from other effects. While we in Paper I only subtracted the pipe draining effect, row one and three in Table 5 now produce a rudimentary drilling mode classification. Here, the output of the physical model observer O1 is used both as an observer output and as a “virtual sensor“ which can serve as input for our AI observers.
This cocept is generalized in Figure 10. In this figure the physical model is represented by one observer for each model output. The AI observers only treat the model as extra system output.

[4] A combination of kick and pump start could in theory give this combination of signatures

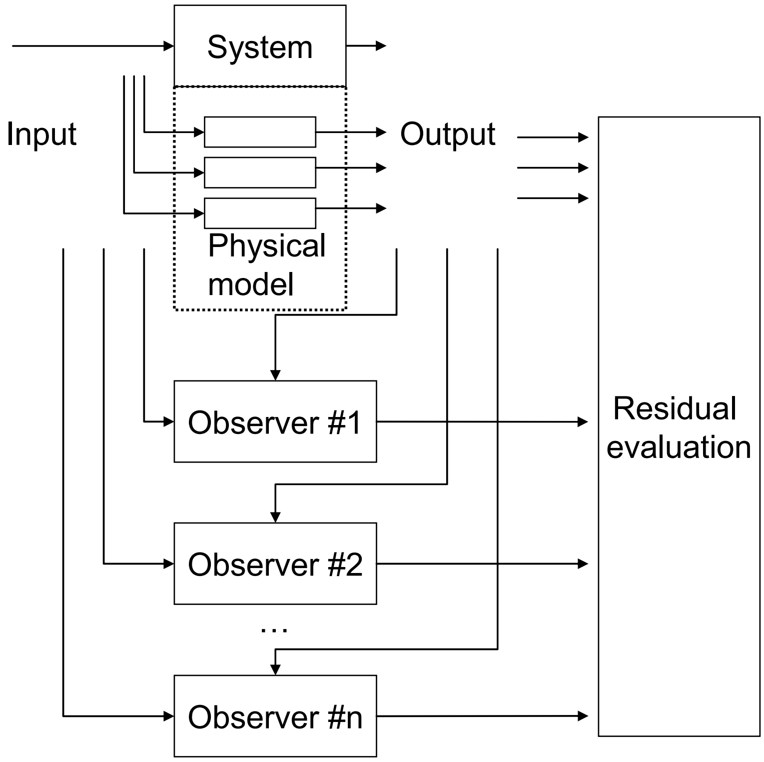

Figure 10: Observers and residual generation when combined with the physical model. The figure reflects that the physical model is not using output feedback at the time of writing.

The e-drilling physical model is not limited to predicting sensor outputs. It can also act as a virtual sensor in the usual sense, for instance by predicting the pressure or temperature at a point in the well where no sensor is placed.
In the next section, we will conclude our discussion on ensembles by presenting methods from a problem-domain which most closely resembles our own.

9.4 One-class classifiers

The direct application of ECOC to the structured residuals that we performed earlier, while close to the original GOS approach, had some shortcomings. Not least with regard to the monolithic matrix which had all observers entering equally in each fault signature, raising issues of how well the approach would scale. Another less visible problem is that we are assuming that all fault signatures will be listed in the matrix. Similarly, most AI classifiers assume that the classes to be distinguished are present in the training set. Both the class imbalance of our examples and the unbounded true model discussed in chapter 6.4 seem to indicate that this assumption is in fact not true for drilling problems in general.

A classification problem where classes not seen in the training set may be encountered, is referred to as *open set classification* (Gori and Scarselli, 1998) and detecting previously unseen behaviour falls under the topic of anomaly or outlier detection. This is also referred to as *one-class classification* and is especially relevant for binary classifiers with a high class imbalance, where the goal is reduced to telling apart one well-documented class of objects from all other seen and unseen classes. The AI observers we presented earlier can be re-interpreted as such anomaly detectors. Ideally, the AI will produce a low residual when the time series falls within the class or classes of behaviour seen in its training set and produce a high residual when presented with new classes of behaviour.

A problem domain with surprising parallels to drilling is intrusion detection in computer networks. As pointed out by (Giacinto et al., 2008), the class of intrusions to be detected is an open set as new exploits are continually being produced. This problem domain is also one with a high class-imbalance as most network traffic is legitimate. Also, most data remain unlabeled. Not only is labeling of traffic data time-consuming, but the set of normal traffic can not be guaranteed to not include an overlooked hacking attempt. As we have seen, these problems are mirrored in the drilling domain. The response of the intrusion detection community has been to develop unlabeled or unsupervised anomaly detection, though these have a higher false alarm rate. A weakness in this approach is that normal behaviour is composed of many different behaviours which are hard to fit within one “normal behaviour” class. We find an echo of this in the naïve structured residuals proposed earlier, which assumed only one signature for the no-fault case. Giacinto (2008) and others have therefore turned to ensembles of one-class classifiers. The approach of Giacinto (2008) is to produce one classifier for each service on the network. A parallel for drilling would be to have one classifier for each non-faulty drilling mode.
As pointed out by (Ding, 2008), research on residual evaluation and threshold computation have received little attention in the FDI community. In Paper IV we saw that a hand-crafted evaluation and threshold function was necessary to detect stick-slip in the torque signal, a situation that is likely to be repeated for other drilling problems. In this regard, the value of introducing observers as anomaly-detectors would be a much more thorough and varied analysis of residual evaluation and thresholding. If we are to expand our reach to new or unseen faults, anomaly-detection offers valuable additions to our toolbox.

9.4.1 Kick detectors as anomaly detectors

In chapter 7 we presented existing kick-detection systems. These kick-detectors are in essence anomaly or one-class detectors. This can be seen from the fact that most novel behaviour will be classified as a kick. We also see that CUSUM relies on noise statistics for the fault-free case, thus focusing on the training examples of one class. The strengths and weaknesses of the alarm systems are thus essentially the same as for one-class detectors (Giacinto et al., 2008), namely robustness in the face of unseen phenomena but also a high number of false alarms. A sketch that illustrates the problem is shown in Figure 11.

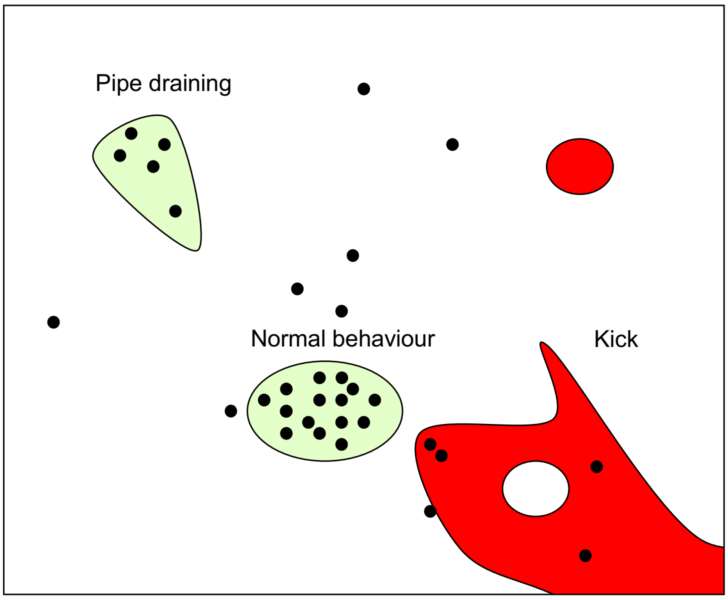

Figure 11: Illustrating classifiers in input space. Black dots indicate examples. The classes are small changes in pit volume (centre), pipe draining (top left) and kick (top and bottom right).

The centre oval in Figure 11 represents the traditional system, which raises an alarm whenever the pit volume increases or decreases significantly, in other words when the example lies outside of the border of the oval. As a one-class or anomaly classifier, this is very robust in the sense that we do not need a large number of kick examples to specify it. However it produces false alarms on safe events such as pipe draining (top left), an effect we explained in chapter 2.

As we have discussed earlier, it is sometimes advantageous to combine several one-class classifiers. As we have also touched on in this dissertation, it is difficult to build a classifier or predictor that relies directly on fault examples in the training set. On the other hand, we have ample opportunity to combine predictors and classifiers to correctly classify more of the safe events, since there are plenty of examples of many safe states. Hence, the data allow us to reduce the false alarm rate.

Reducing the number of false alarms is a goal in itself when it comes to improving security in drilling operations (Heber and Åsland, 2007), but it also carries an additional bonus. The sensitivity of today's kick detection systems is typically given by a threshold value for the deviation between measured and predicted pit volume. This threshold value is set so that it strikes a balance between detecting a kick early and keeping the number of false alarms at a tolerable level. If we can independently weed out some of the false alarms, the driller can increase the sensitivity of the kick detector and get an even earlier warning of kicks.

By this strategy, we are thus able to reduce both type I and type II errors in the alarm system without presenting the AI with kick examples, circumventing a deficiency in the data set. In chapter 10.1 we present our experiments using this strategy.

10 Experimental results

In this chapter we summarize our experiments and findings, both those presented in more detail in the papers and complementary experiments.

10.1 Increasing the sensitivity of the kick-detection system

In chapter 9.4 we discussed one-class detectors and laid out a strategy for improving kick detection using supervised learning, while handling the class imbalance. Here we summarize our experiments using this strategy, which also includes a grey-box approach.

10.1.1 Experiments

In Paper I and Paper II we presented an experiment which is a proof-of-concept of this strategy and in this section we briefly review the results.

In Paper I we implemented the greybox approach when we corrected for the pipe draining. We showed that while the AI alone was not able to make a better prediction than the physical model, the greybox approach reduced the false alarm rate by around 25% compared to the physical model. The AI used pump rate and measured mud density of the outflow to predict flow rate. As a bonus, the mud density also allowed the AI to correct for a weakness in the physical model that we were at first unaware of. We take this as a proof-of-concept that the greybox approach can be used to correct for different modelling errors and through that reduce the number of false alarms.

When the physical model was later corrected, the mud density readings were no longer needed. Thus in Paper II we only used pump rate as an input. In light of the curse of dimensionality, this was important for reliability. It would be easier to test the behaviour of the AI under all conceivable circumstances if there were few inputs and the behaviour of the pump signal is well understood.
It was also crucial that the AI did not learn a kick signature. We could not guarantee that the time series did not include small fluid kicks or examples of wellbore breathing. It is not inconceivable that at least the latter could have been learned by the AI, for instance through mud density or bottom hole temperature. This would hide the wellbore breathing from the operator, masking it as pipe effects.
Limiting the AI to learn from the pump rate avoids this. While a change in pump rate may cause a kick or wellbore breathing, these examples will belong to the minority in most time series and predicting "no kick during pump stop" is the optimal choice for any learning algorithm. Conversely, a kick causes no change in the pump rate. We can therefore guarantee that the AI will not learn and mask kicks even if its training set contained a few examples of it. We have met this consideration before, in the robustness criteria of the observer. The pump rate is best understood as a system input and our AI can therefore be seen to have no output feedback. This makes it robust to all faults except that of the pump sensor, but also means that it will be prone to drifting. This is however not a problem for pipe draining, which takes place on a time scale too short for the drifting to have an effect.

In Paper II we focused on pump stop events instead of time samples, and counted the percentage of pump stops that gave a false alarm, believing this to be an alarm rate

measure more relevant to the driller. Focusing only on the first 15 minutes after a pump stop, we did not use the greybox approach. None the less we achieved good results, as many of the effects modelled by the physical model were not taking place at these times. Well A contained 34 examples, of which two thirds were used for training and 10 for performance testing. Well B drilled from the same rig contained 131 examples of which all were used for performance testing.
We found that the false alarm rate was reduced from near 100% to 47% and 26% at 3bbl sensitivity for well A and B respectively.

From the above results it appears that the trained AI were able to generalise to new wells. Though the false alarm rate may still be too high for real-world use, the AI delivers a real improvement on flow predictions. The fact that it actually performs better on the unseen well may be a coincidence, but we also speculate that well A contained examples of wellbore breathing so that some of the examples were in fact true alarms. If this was the case, it lends support to our claim that the AI in this case could learn from a data set with a few mislabeled examples.

10.1.2 Evaluating the choice of AI

For the AIs in Paper I and Paper II, we chose the Echo State Network (ESN) (Jaeger, 2001, Jaeger, 2002) and it was our starting hypothesis that the ESN had several characteristics that made it especially well suited to this task. First of all, the ESN does not require that we specify the length of the input window, which is an advantage when the optimal window length is not known. Secondly, a pipe or a tank which drains is a dynamic system which is easily modelled by a few feedback loops. Since the ESN consists of feedback loops, we believed that pipe draining would have a particularly simple representation inside ESNs and therefore be easier to learn for an ESN than for AIs with no recurrent couplings.

We compared the ENSs performance against the well-known Autoregressive Moving Average (ARMA) (Theodoridis and Koutroumbas, 2006) and found that the ESN did indeed outperform ARMA. However, as is often the case when new methods are compared against old ones, the new method had more effort invested in its implementation. We can not rule out that the performance of ARMA or a similar method could have been improved with more careful tuning. Neither is it clear if ARMA was more adversely affected by the possible wellbore breathing in well A. Our results are therefore inconclusive on the question of ESN as a superior or inferior choice.

10.2 Stuck pipe detection using statistical features

In Paper IV we presented a statistical method for stuck pipe prediction which has been studied at SINTEF and as a part of this dissertation. This diagnosis seeks to detect poor hole cleaning by constructing statistical features from bottom hole pressure (BHP) and torque (TRQ) signals.

10.2.1 Summary of the method

For a window of n samples $X = \{x_1, ..., x_n\}$, their average μ and standard deviation σ, the skew is defined as:

$$\text{skew} \equiv \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right)^{3/2}} = \frac{\mu_3}{\sigma^3} \tag{10.1}$$

The normalized standard deviation is a dimensionless quantity defined as:

$$\sigma_n \equiv \frac{\sigma}{\mu} \tag{10.2}$$

We build on previous work by (Jardine et al., 1995, Rezmer-Cooper, 2002) when we propose the diagnostic signal

$$F \equiv skew(BHP) * \sigma_n(TRQ) \tag{10.3}$$

However, in our analysis of F we depart from the methods Jardine (1995) and Rezmer-Cooper (2002). They suggested integrating the diagnostic signal over time and thresholding the resulting values. It is difficult to separate poor hole cleaning from harmless deviations using this scheme, which in turn leads to many false alarms. We observe that for normal conditions, the diagnostic signal will fluctuate around zero but during poor hole cleaning will show spikes with a positive sign. In our implementation, F is computed for every n sample. After w calculations of the diagnostic, (for $n*w$ samples) we compute the percentage of positive diagnostic signals R and note the maximum value $P_{\max}$ of the positive signals. We then raise an alarm if these values exceed given thresholds:

$$R > Th \text{ and } P_{\max} > Th_{\max} \tag{10.4}$$

10.2.2 Experiments

As detailed in Paper IV, real-world data indicate that the percentage of positive spikes is a more robust indicator than the integrated diagnostic signal. We also find that the alarms produced by our method correlate well with an existing alarm system in the e-drilling model, which works by predicting cutting concentrations along the well. This served as a reality-check on the link between hole cleaning and our diagnostic signal. We also had the opportunity to test the diagnostic signal against recent poor hole cleaning and stuck pipe events. Figure 12 shows the bit depth in a north-sea well over 48 hours. A period of poor hole cleaning occurred at 15-20 hours and a stuck pipe incident at 26 hours. In both cases we see that the bit was eventually pulled off bottom to perform a hole-cleaning procedure, in response to the problems. In Figure 13 we see the diagnostic signal and warnings produced over these 48 hours. For the first problem, our method seems to detect poor hole cleaning as it occurs. For the stuck pipe incident, we find that the alarm is raised approximately 23 minutes *before* the stuck pipe incident. The proposed method could therefore in this case have provided an early warning of stuck pipe.

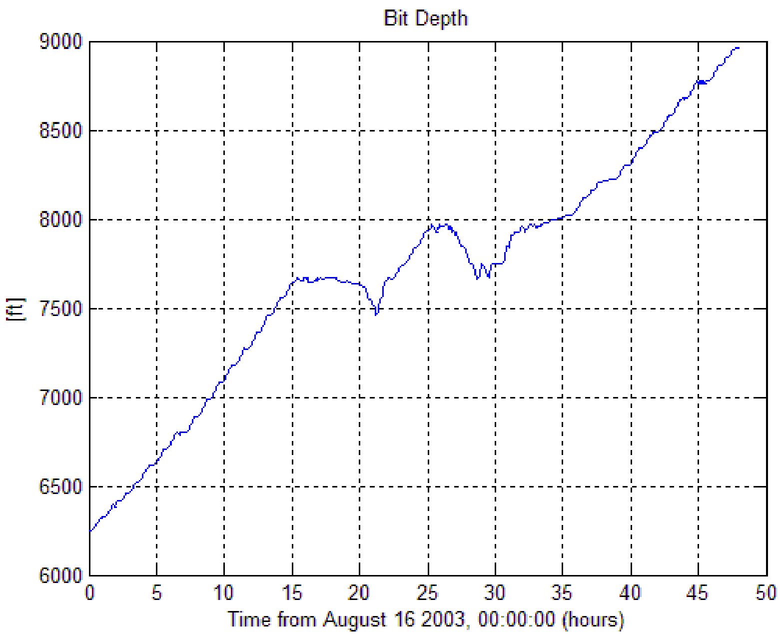

Figure 12: Bit depth over a 48 hour period. Poor hole cleaning is reported at 15-20 hours and a stuck pipe is reported at about 26 hours.

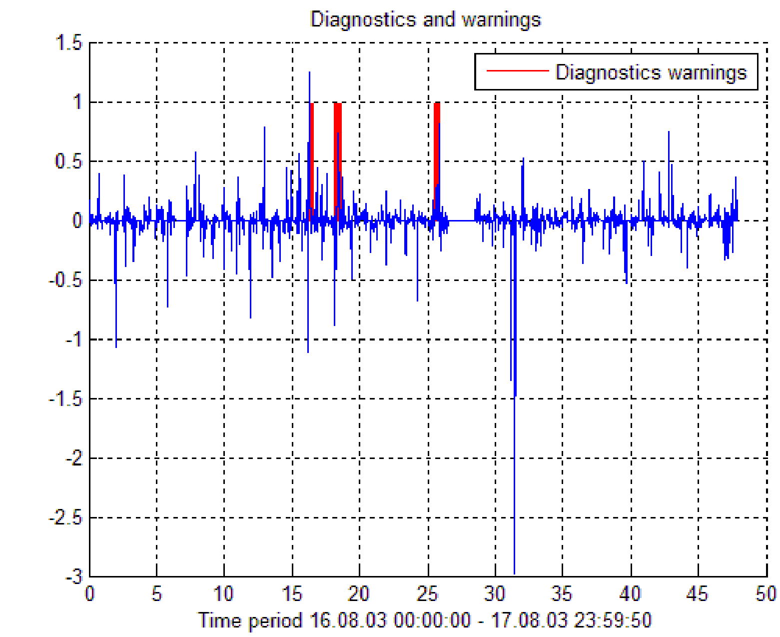

Figure 13: Diagnostic signal and warnings running in parallel with Figure 12. Both problems are detected, with no false alarms in this time series. $w = 8$, $Th = 0.85$ **and** $Th_{\max} = 0.2$

In time series from a separate well, a drill string twist-off due to drill string wash-out was recorded and we had the opportunity to observe how our diagnostic signal behaved in this case. This is shown in Figure 14. The twist-off occurred at around the 400 point and warnings were generated more than 3 hours in advance. Different parameters of the diagnostics could have given even earlier warnings, but the method does not appear dependent on fine-tuning of the parameters for early detection.

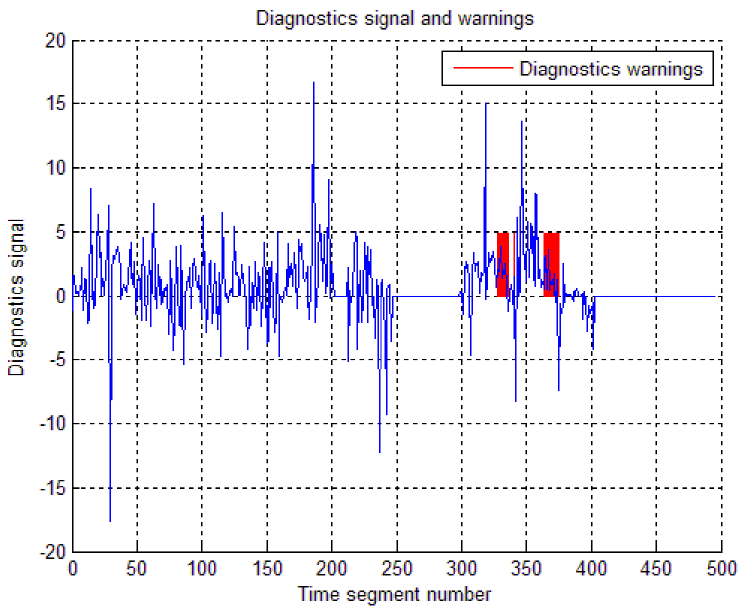

Figure 14: Value of diagnostic signal with a 60 minutes window, $Th = 1$ and $Th_{\max} = 0$ with alarms in red. Drill string twist-off occurred at around 400. The first warning is generated about 3 hours and 20 minutes before the twist off.

10.2.3 Sufficiency of data

The approach of Paper IV exemplifies the remedies to the raw data problem we described in chapter 6.2. The algorithm is based on expert knowledge, employs near-faults as well as faults, and has only four free parameters: n, w, Th and $Th_{\max}$ of which two enter in the feature dimensionality reduction. We will not attempt to derive bounds on the VC dimension for this algorithm, but we note that the threshold functions in equation (10.4) may be represented by neurons with these threshold functions. Also the ratio R may be similarly constructed, implying that a neural network with only 3-4 free parameters could have accomplished the classification. We are therefore led to a VC dimension of 4 in this equivalent problem. This in turn implies that around 40 examples is a minimum for good generalisation performance. In our paper, $n * w$ samples could typically span an hour. Two days of varied examples from drilling would then satisfy the minimum number of examples.

Available time series therefore offer a solid basis on which to tune, assess and further develop the proposed method

10.3 Improving pipe-draining prediction by ensemble averaging

In chapter 10.1 we trained several ESNs and selected the one with the best performance. Here we explore the hypothesis that that method could be improved if we instead employed ensemble methods as discussed in chapter 3.4 and 9.

We start by training 50 randomly initialized echo state networks using the first 60 percent of the well A data as a training set. Each ESN is initialized with between 150 and 250 nodes and trained on a randomly chosen but contiguous subset of the training data.

In Figure 15 we see a typical behaviour of an ESN on two hours of the test-set data. It predicts the first two pump drainings fairly well except for a spike, but shows erratic behaviour about halfway. This can be compared with Figure 16 and Figure 17 showing the performance when taking the average or median of the 50 ESNs.

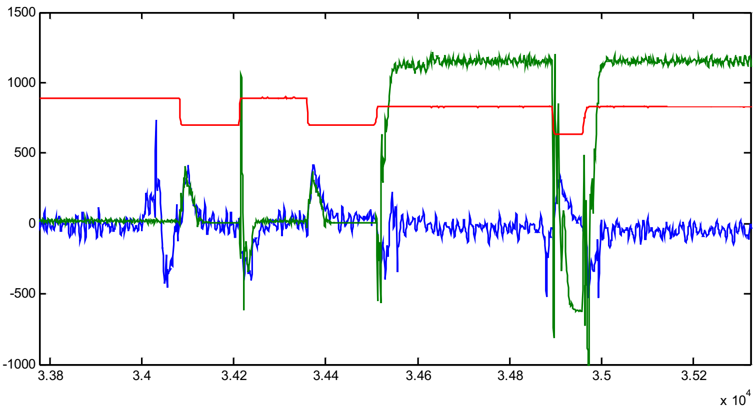

Figure 15: Residual flow (blue), prediction by a single ESN (green), pump rates not to scale (red). Prediction and residual flow have been synced at the start of the plot

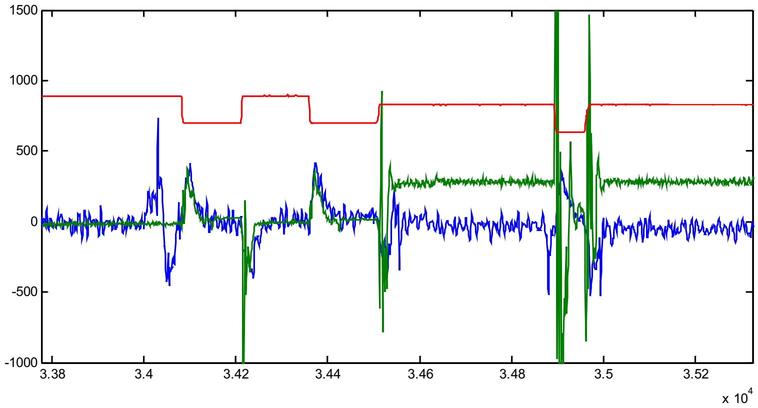

Figure 16: Same plot as in Figure 15 but using the average of the 50 ESNs

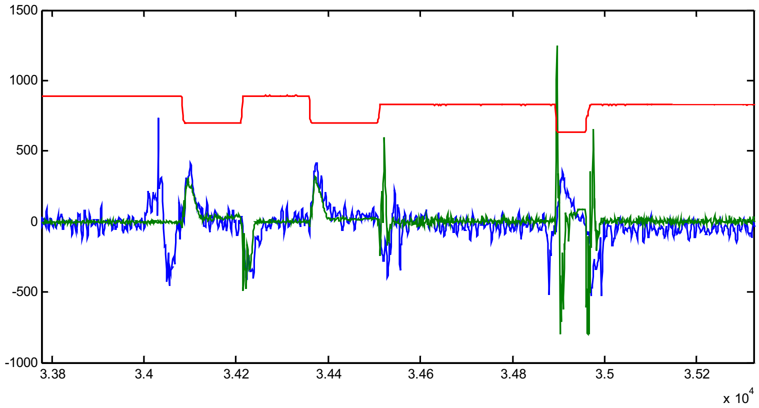

Figure 17: Same plot as in Figure 15 but using the median of the 50 ESNs

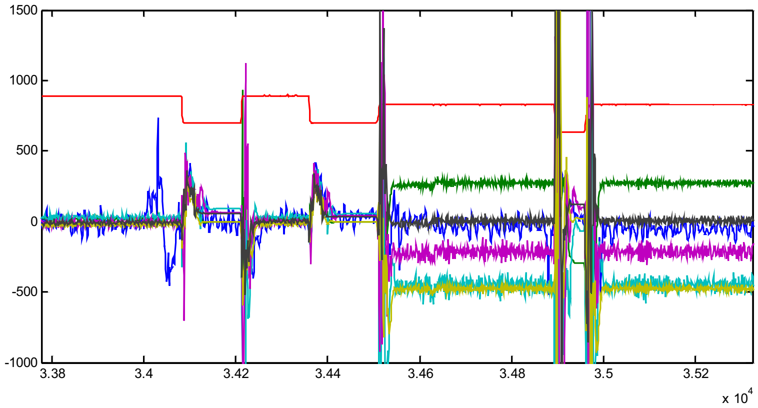

Figure 18: The individual predictions by five of the ESNs

We find that the ensemble approach using the median removes noise such as spikes or sudden jumps from the predictions. This can be seen from Figure 18 which shows how five different ESN predictions compare. We see that the predictions tend to agree when making correct predictions, while the errors in the seemingly difficult parts tend in no particular direction, leading to an average close to the true value. This seems to agree well with ensemble theory, which states that the predictions are improved by reducing the variance.

In our case, median seems to outperform average. This can be understood as pathological outliers having a disproportionate effect on the average. We should also compare median against the strategy of choosing the ESN that performs best on the training set. We find that the best single ESN judged by its training set performance, has a false alarm rate of 32% on the training set and 60% on the test set. The ensemble has a false alarm rate of 37% on the training set and 14% on the test set.
We find that the ensemble performs better than the best member and also that it outperforms the best ESN in Paper II. It should however be mentioned that the ensemble seem to make little or no improvement in the parts of the time series that are already predicted fairly well by individual ESNs.

In a real-world setting, where plots of the predictions may be analyzed by a human in real-time, reliability is perhaps more important than accuracy. The greatest contribution of the ensemble is not a closer fit to the pipe draining curve, but that the ensemble reduce the number of complete failures in the prediction. Reducing the number of such “breakdowns” is crucial if predictive AIs are to be integrated into the data-flow. On the other hand, these failures could be a characteristic of echo state networks or the fact that the observer has no output feedback. Until these issues are settled, the utility of the ensemble approach should be judged on a case-by-case basis.

11 Concluding remarks

Calls for improvement in machine learning and data mining often fall into one of two camps: Either we need more data or we need better algorithms. Even as the data rate from drilling operations is making large leaps in terms of bandwidth, the number of examples of drilling problems is luckily not set to increase in the same fashion. In this dissertation we have taken the view that training set size will not increase by leaps and bounds in the future. The training set will continue to show noise, be class-imbalanced, mostly unlabeled and drilling problems will remain an open set classification problem even with more data. We therefore find ourselves at the latter camp, calling for better algorithms and smarter architectures.

11.1 Conclusions

With the above viewpoints in mind, we have pursued two approaches: Greybox models and ensemble learning. We have shown that the greybox approach succeeds at simplifying the learning task, allowing us to easily implement AIs that quickly bring added value to prediction tasks and alarm systems. The greybox approach is therefore an attractive strategy for bringing machine learning technology to market.

We have also demonstrated that ensemble learning can improve the predictions of our AIs and that the problems of unlabeled data and class imbalance can be circumvented. The problem of kick-detection suffers from the latter two problems. We have shown that when traditional kick alarm systems and AIs are combined, there exists a machine learning strategy that leads to an improved alarm system, even in the absence of known kick examples in the training set. This is a strategy that takes into account both the scarcity of relevant fault examples and the high cost of manual labelling of drilling time series.

Furthermore we have presented a stuck pipe detection algorithm demonstrating the value of analysing the “noisy” components in the time series.

The greybox approach used in our studies combined the best from machine learning and physical models. In line with this, we have investigated architectures that combine techniques from traditional fault detection and isolation (FDI) with methods from machine learning. It is our impression, as outsiders, that the field of FDI has reached an impasse in recent years. Results from linear FDI are unified and firmly established, but approaches that reflect the nonlinearity of the system under supervision deal with models of only limited complexity and advances are tied to specific models. Machine learning is acknowledged as a way forward for FDI (Patan, 2008) and AIs have been produced that solve fault detection problems. These efforts however, do seemingly not bridge the gap between the FDI field and machine learning by combining experience from the two fields. Instead, generalist machine learning algorithms are imported and applied relatively unchanged. We have proposed architectures that combine machine learning with existing design in FDI. More importantly, we have proposed that ensemble learning theory can be used to analyse observer schemes from a new angle and serve as an interface between FDI and machine learning. This is our contribution to rejuvenating the discussion on machine learning in FDI.

11.2 Further work

Here we summarize avenues of directions that may be pursued further.

- Experiments with our proposed GOS-like architectures should be carried out

- The systems we have experimentally tested should undergo further testing to confirm that they are robust to new situations.

- Some effects that can be learned are specific to a formation, to a rig or could change while drilling. One such case is the pipe-draining signature that could change as the flow is re-routed in the rig's pipe system. Thus it might be beneficial to explore online learning, where the AI is updated on-site. However, this brings in issues of stability and predictable performance which must be addressed. Ensemble learning offers an attractive way to do online learning, where new data is incorporated simply by training new AIs on it, leaving AIs trained on old data unchanged.

- While our end goal was fault detection, the greybox approach on its own produced a more accurate prediction of a key variable, which is valuable in itself. It might be worthwhile to apply this more broadly to correct for calibration errors and improve prediction accuracy.

Bibliography

ABONYI, J., BABUSKA, R., VERBRUGGEN, H. B. & SZEIFERT, F. (2000) Incorporating prior knowledge in fuzzy model identification. *International Journal of Systems Science,* 31**,** 657-667.

ALCORTA GARCÍA, E. & FRANK, P. M. (1996) Analysis of a class of dedicated observer schemes to sensor fault isolation. *Control '96, UKACC International Conference on (Conf. Publ. No. 427).*

ALCORTA GARCÍA, E. & FRANK, P. M. (1997) Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice,* 5**,** 663-670.

ALDRED, W., PLUMB, D., BRADFORD, I., COOK, J., GHOLKAR, V., COUSINS, L., MINTON, R., FULLER, J. & GORAYA, S., TUCKER, DEAN (1999) Managing drilling risk. *Oilfield review.*

AWASTHI, A., SANKARAN, S., NIKOLAU, M., SAPUTELLI, L. & MIJARES, G. (2008) SPE 111853 - Meeting the Challenges of Real-Time Production Optimization—A Parametric Model-Based Approach. *Intelligent Energy Conference and Exhibition.* Amsterdam, SPE.

BASSEVILLE, M. & BENVENISTE, A. (1986) *Detection of abrupt changes in signals and dynamical systems,* Berlin, Springer.

BESTLE, D. & ZEITZ, M. (1983) Canonical form observer design for non-linear time-variable systems. *International Journal of Control,* 38**,** 419 - 431.

BREIMAN, L. (1996a) Bagging predictors. *Machine Learning,* 24**,** 123-140.

BREIMAN, L. (1996b) Heuristics of instability and stabilization in model selection. *Annals of Statistics,* 24**,** 2350-2383.

BUHLMANN, P. & YU, B. (2002) Analyzing bagging. *Annals of Statistics,* 30**,** 927-961.

BUJA, A. & STUETZLE, W. (2000) The effect of bagging on variance, bias and mean squared error. AT&T Labs Research.

BURNHAM, K. P. & ANDERSON, D. R. (1998) *Model selection and inference: a practical information-theoretic approach*.

CHAVES, M. & SONTAG, E. D. (2002) State-estimators for chemical reaction networks of Feinberg-Horn-Jackson zero deficiency type. *European Journal of Control,* 8**,** 343-359.

CHAWLA, N. V., JAPKOWICZ, N. & KOLCZ, A. (2004) Editorial: special issue on learning from imbalanced data sets. *SIGKDD explorations,* 6**,** 1.

DIETTERICH, T. G. & BAKIRI, G. (1995) Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research,* 2**,** 263-286.

DING, S. X. (2008) *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer.

DOMINGOS, P. (2000) A unified bias-variance decomposition for zero-one and squared loss. *17th National Conference on Artificial Intelligence (AAAI-2000)/12th Conference on Innovative Applications of Artificial Intelligence (IAAI-2000).* Austin, Tx.

EPSIS & ABB (2006) Potential value of integrated operations on the norwegian continental shelf. *Report commisioned by the Norwegian Oil Industry Association (OLF).*

ERDOGMUS, D., GENC, A. U. & PRINCIPE, J. C. (2002) A neural network perspective to extended Luenberger observers. *Measurement & Control,* 35**,** 10-16.
FORSSELL, U. & LINDSKOG, P. (1997) Combining semi-physical and neural network modeling: an example of its usefulness. *11th IFAC Symposium on System Identification (SYSID'97).*
FORTUNA, L., GRAZIANI, S., RIZZO, A. & XIBILIA, M. G. (2007) *Soft Sensors for Monitoring and Control of Industrial Processes*, Springer.
FRANK, P. M. (1990) Fault-diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results. *Automatica,* 26**,** 459-474.
FRANK, P. M., DING, S. X. & MARCU, T. (2000) Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control,* 22**,** 57-101.
FRANK, P. M. & DING, X. C. (1994) Frequency-domain approach to optimally robust residual generation and evaluation for model-based fault-diagnosis. *Automatica,* 30**,** 789-804.
FRIEDMAN, J. H. & HALL, P. (2007) On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference,* 137**,** 669-683.
FRUHWIRTH, R. K., THONHAUSER, G. & MATHIS, W. (2006) SPE 103217, Hybrid simulation using Neural Networks to Predict Drilling Hydraulics in Real Time.
GARCÌA, E. A. & FRANK, P. M. (1997) Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice,* 5**,** 663-670.
GAYAKA, S., YAO, B. & MECKL, P. (2007) Fault Detection for Nonlinear Systems in Presence of Input Unmodeled Dynamics. *2007 ieee/asme international conference on advanced intelligent mechatronics.*
GIACINTO, G., PERDISCI, R., DEL RIO, M. & ROLI, F. (2008) Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion,* 9**,** 69-82.
GORI, M. & SCARSELLI, F. (1998) Are multilayer perceptrons adequate for pattern recognition and verification? *Ieee Transactions on Pattern Analysis and Machine Intelligence,* 20**,** 1121-1132.
HAND, D. J., MANNILA, H. & SMYTH, P. (2001) *Principles of data mining,* Cambridge, Mass., MIT Press.
HANSEN, L. K. & SALAMON, P. (1990) Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12**,** 993-1001.
HARGREAVES, D., JARDINE, S. & JEFFRYES, B. (2001) SPE 71369 - Early Kick Detection for Deepwater Drilling: New Probabilistic Methods Applied in the Field. *SPE Annual Technical Conference and Exhibition.* New Orleans.
HEBER, H. & ÅSLAND, J. E. (2007) Human factors i bore og brønnoperasjoner - Borernes arbeidssituasjon.
HERNANDEZ, J. L., BISCAY, R., JIMENEZ, J. C., VALDES, P. & DEPERALTA, R. G. (1995) Measuring the dissimilarity between EEG recordings through a nonlinear dynamical system approach. *International Journal of Bio-Medical Computing,* 38**,** 121-129.
HO, T. K. (1998) The random subspace method for constructing decision forests. *Ieee Transactions on Pattern Analysis and Machine Intelligence,* 20**,** 832-844.

HO, T. K. (2000) Complexity of classification problems and comparative advantages of combined classifiers. IN KITTLER, J. & ROLI, F. (Eds.) *1st International Workshop on Multiple Classifier Systems (MCS 2000).* Cagliari, Italy.

IOANNIDIS, J. P. A. (2005) Why most published research findings are false. *Plos Medicine,* 2**,** 696-701.

JAEGER, H. (2001) The "echo state" approach to analyzing and training recurrent neural networks. GMD Report 148, German National Research Institute for Computer Science.

JAEGER, H. (2002) A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. Fraunhofer Institute for Autonomous Intelligent Systems (AIS).

JARDINE, S. I., MCCANN, D. P. J. & WHITE, D. B. (1995) Method of warning of pipe sticking during drilling operation.

KONG, E. B. & DIETTERICH, T. G. (1995) Error-correcting output coding corrects bias and variance. *Twelfth International Conference on Machine Learning.*

LAUER, F. & BLOCH, G. (2008) Incorporating prior knowledge in support vector regression. *Machine Learning,* 70**,** 89-118.

MAGNI, J. F. & MOUYON, P. (1994) On residual generation by observer and parity space approaches. *Automatic Control, IEEE Transactions on,* 39**,** 441-447.

MARCU, T., KÖPPEN-SELIGER, B., FRANK, P. M. & DING, S. X. (2003) Dynamic functional-link neural networks genetically evolved applied to fault diagnosis. *European Control Conference (ECC).* Cambridge, UK.

NORTVEDT, R., SVENSKA KEMISTSAMFUNDET & NORSK KJEMISK SELSKAP . FAGGRUPPE FOR KJEMOMETRI (1996) *Anvendelse av kjemometri innen forskning og industri,* [Oslo], Tidsskriftforlaget Kjemi.

PALIT, A. K. & POPOVIĆ, D. (2005) *Computational intelligence in time series forecasting: theory and engineering applications,* London, Springer.

PAPATHANASSIOU, C. & PETROU, M. (2002) Incorporating prior knowledge in ICA. IN SKODRAS, A. N. C. A. G. (Ed.) *14th International Conference on Digital Signal Processing (DSP 2002).* Santorini, GREECE, Ieee.

PATAN, K. (2008) Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes. *Lecture Notes in Control and Information Sciences 377.* Berlin, Heidelberg, Springer-Verlag Berlin Heidelberg.

PETERSEN, J., BJØRKEVOLL, K. S., FRØYEN, J. & ROMMETVEIT, R. (2006) IBP1373_06 A General Dynamic Model For Flow Related Operations During Drilling, Completion, Well Control and Intervention. *Rio Oil and Gas Expo and Conference 2006.*

POLIKAR, R. (2007) Bootstrap-inspired techniques in computational intelligence. *Ieee Signal Processing Magazine,* 24**,** 59-72.

POLIKAR, R. (2008) Ensemble learning. *Scholarpedia.*

RAJAGOPAL, R. & POTTER, L. C. (2003) Multivariate MIMO FIR inverses. *Image Processing, IEEE Transactions on,* 12**,** 458-465.

RASMUSSEN, J. (1986) Information Processing and Human-Machine Interaction, An Approach to Cognitive Engineering.

RENARDY, M. & ROGERS, R. C. (2004) *An introduction to partial differential equations,* New York, Springer.

REZMER-COOPER, I. (2002) Method for detecting stuck pipe or poor hole-cleaning.

SAPUTELLI, L., MALKI, H., CANELON, J. & NIKOLAOU, M. (2002) SPE 77703-MS, A Critical Overview of Artificial Neural Network Applications in the Context of Continuous Oil Field Optimization. *SPE Annual Technical Conference and Exhibition.* San Antonio, Texas.

SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. & LEE, W. S. (1998) Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics,* 26**,** 1651-1686.

SCHREIBER, T. (1997) Detecting and analyzing nonstationarity in a time series using nonlinear cross predictions. *Physical Review Letters,* 78**,** 843-846.

SCHREIBER, T. & SCHMITZ, A. (1997) Classification of time series data with nonlinear similarity measures. *Physical Review Letters,* 79**,** 1475-1478.

SHERIDAN, T. B. (1987) Supervisory Control. IN SALVENDY, G. (Ed.) *Handbook of Human Factors.*

SKALLE, P. & AAMODT, A. (2005) Knowledge-Based Decision Support in Oil Well Drilling. *Intelligent Information Processing II.*

STAROSWIECKI, M. & COMTET-VARGA, G. (2001) Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica,* 37**,** 687-699.

TANAKA, M., SHIROMARU, I., SAKAWA, M., MATSUMOTO, T. & IEEE (1995) Application of Kohonen's self-organizing network to the diagnosis system for rotating machinery. *1995 IEEE International Conference on Systems, Man and Cybernetics - Intelligent Systems for the 21st-Century.* Vancouver, Canada, I E E E.

TERRA, M. H. & TINOS, R. (2001) Fault detection and isolation in robotic manipulators via neural networks: A comparison among three architectures for residual analysis. *Journal of Robotic Systems,* 18**,** 357-374.

THEODORIDIS, S. & KOUTROUMBAS, K. (2006) *Pattern recognition,* Amsterdam ; Boston, Elsevier/Academic Press.

THONHAUSER, G. (2004) Using Real-Time Data for Automated Drilling Performance Analysis. *Oil Gas European Magazine.*

THONHAUSER, G. & MATHIS, W. (2006) SPE 103211, Automated Reporting Using Rig Sensor Data Enables Superior Drilling Project Management.

VALENTINI, G. (2003) Ensemble methods based on bias–variance analysis. *Dipartimento di Informatica e Scienze dell'Informazione.* Genova, DISI, Universit`a di Genova.

VERLEYSEN, M. (2003) Learning high-dimensional data. *Limitations and Future Trends in Neural Computation, IOS Press***,** 141-162.

WATSON, D., BRITTENHAM, T., MOORE, P. L. & HENRY L. DOHERTY MEMORIAL FUND OF AIME (2003) *Advanced well control,* Richardson, Tex., Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers.

WESTPHAL, L. C. (2001) *Handbook of Control Systems Engineering.*

WOLPERT, D. H. (1996) The lack of A priori distinctions between learning algorithms. *Neural Computation,* 8**,** 1341-1390.

YU, D. L., GOMM, J. B. & WILLIAMS, D. (1999) Sensor fault diagnosis in a chemical process via RBF neural networks. *Control Engineering Practice,* 7**,** 49-55.

ZAK, M., PARK, H. & IEEE, I. (2001) Gray-box approach to fault diagnosis of dynamical systems. *IEEE Aerospace Conference.* Big Sky, Mt, Ieee.

ZHANG, X. D., POLYCARPOU, M. M. & PARISINI, T. (2008) Design and analysis of a fault isolation scheme for a class of uncertain nonlinear systems. *Annual Reviews in Control,* 32**,** 107-121.
ZHAO, S. H. & DILLON, T. S. (1997) Making neural networks more intelligible by incorporating prior knowledge. *1997 International Conference on Neural Information Processing and Intelligent Information Systems.* Springer-Verlag.

Papers

Paper I

Is not included due to copyright

Paper II

Is not included due to copyright

Paper III

Notice: An extended and revised version of Paper III has been submitted to the journal Neurocomputing, as part of a special issue of papers from the ESTSP 08 conference.

Time series opportunities in the petroleum industry

Roar Nybø *

SINTEF Petroleum Research
Thormøhlensgate 55, 5008 Bergen - Norway

Abstract. Soft computing techniques have gained greater interest and acceptance in the oil industry in recent years. Some, who advocate the education of more interdisciplinary petroleum engineers, even list soft computing as one of the core competencies for such engineers. This paper will give a brief introduction to the challenges and opportunities for applied time series prediction in the oil industry and recent trends in research, with a focus on fault prediction.

1 Introduction

The petroleum industry, while traditionally conservative, has a surprisingly long history of testing and deploying artificial intelligence (AI) or soft computing systems. Early examples include expert systems like “Prospector” from the late 70’s for evaluating mineral deposits and “Dipmeter advisor” from the 80’s [1], which dealt with inferring 3D geological structures from measurements taken along the borehole. The early 90’s saw the commercial launch of “ODDA”, an expert system advisor for directional drilling developed by Total and Norsk Hydro [2] and the “Analysis While Drilling” package developed by Total and Nordic Offshore Systems [3].

When an oil well is drilled, equipment failure or a misjudgement of downhole conditions may delay the operation by days or weeks. One need only consider the cost of renting a drilling rig, now exceeding half a million dollars per day, to see that the cost of faults may easily enter the million dollar range. These high stakes increase the risk or perceived risk of trying out unproven technology, partly explaining the conservative attitude [4]. On the other hand, the drilling contractor would get an immediate return on their investments in fault prediction software even if it delivered only a small increase in the ability to predict and avoid faults. Thus ideally, a fault prediction system could be developed incrementally and still be useful and justify industry support in its early stages.

This paper seeks to give an overview of recent developments in the petroleum industry, its use of time series prediction methods as well as the characteristics of its time series, research challenges, open problems and possible development.

* This work is funded in part by the "Center for Integrated Operations in the Petroleum Industry" (http://www.ntnu.no/iocenter)

2 Integrated operations

Currently AI or "soft computing" methods are finding increased acceptance as one of the tools for deploying "Integrated Operations" (IO). Also known variously as e-operations and digital oil fields, the term loosely encompass a move for cutting costs and increasing oil recovery using new computer technology. Some broad themes can be outlined. One is how the oil industry is importing ideas from the process industries, such as a tighter integration between the oil companies and their suppliers when it comes to logistics and project management, as well as analyzing and optimizing offshore oil platform performance on the same terms as for a factory.

Another eye-catching feature of IO is the use of extensive video-conferencing between on- and offshore facilities and 3D visualization of the oil field and ongoing well drilling [5]. This has the aim of integrating different disciplines into planning and real-time operations. It also advances the industry's goal of keeping more of their personnel in onshore offices, being available for consultation with several platforms.

Of most interest may be the increase in real-time data that the oil industry has seen in recent years. This is mainly due to new downhole measurement equipment and an increase in bandwidth between this equipment and the offshore rig [6, 7] as well as the rig and land based facilities. Much of the ongoing research in IO seeks to take advantage of this torrent of data. Efforts include real-time production optimization [8] detailed monitoring of fluid flow [9] and adjusting the path of a well during drilling, based on real-time downhole surveys of the rock formation. While such real-time measurements have been available for years, their bandwidth was previously limited to around 20 bits/sec [10]. Challenging optimization problems also abound in the area of time series data analysis, such as predicting the interactions between a large number of wells in order to optimize their total production.

All this has created a need for a stronger ICT-literacy in the oil industry, where people such as Prof. Ershaghi at the Center for Interactive Smart Oilfield Technologies † at U. of Southern California are among the ones arguing for a revision of the petroleum engineering education, with data mining and soft computing as two of the core competencies.

3 Properties of oil industry time series

Time series in the oil industry are of course generated from a multitude of different processes, but a short overview may still give a feel of how it differs from the textbook examples of time series. Asking an industry professional about the series most prominent feature, the answer is likely to be "*noise*". Grave inaccuracies in the measurements contribute substantially, but "noise" may also be aspects of the system not covered by our models. For instance, the drillstring (Figure 1), as any rotating equipment, may fall prone to vibrations and wobbling. This may affect not just measurements of the drillstring's torque and weight, but also fluid flow and pressure [11]. The drillstring, several kilometres long, may in turn have had its movement affected by the type and amount of gravel in the well.

† http://cisoft.usc.edu/

This messy and very much “real world” interconnectedness of different processes has long been acknowledged as a challenge for traditional models [3]. However, it also lets a feature such as wobbling make its fingerprint on many variables. It is enticing that this correlation may let a multivariate analysis extract early warning signs from what is generally regarded as noise.

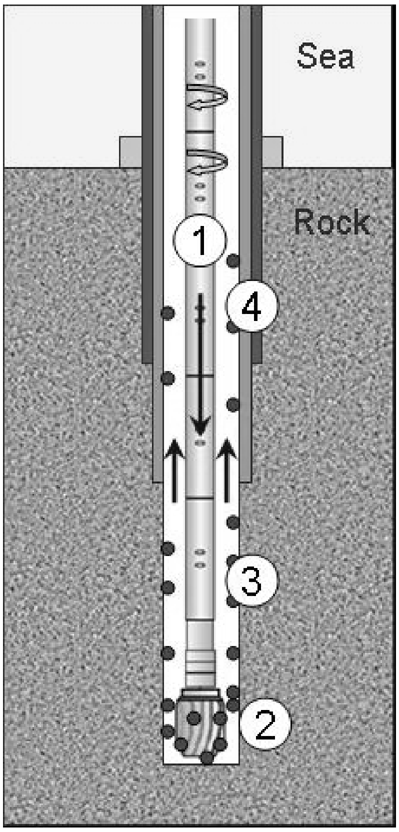

Figure 1: Simplified schematics of oil well drilling. A rotating pipe (1) extends from the rig to the bottom of the well, where it drives a drillbit. (2) At the same time fluid is being pumped down the pipe. This returns to the rig along the outside of the pipe, carrying the crushed rock (3) along with it. As drilling progresses, the wall of the well is periodically fitted with a protective casing (4).

3.1 Pre-processing and problem definition

For the purpose of downhole monitoring, our task can often be framed as that of an inverse problem: Given our measurements, reconstruct the downhole conditions that caused them. Measurements of rock formation properties are coarse and real-time measurements along the well are sparse with current technology, frequently making the inverse problem an ill-posed one [12].

Fault detection and prevention may also be framed as a time-series prediction problem: Given the time-series up to now, predict if a fault is likely to occur. The horizon of such a task will be problem-specific. While the first signs of gas having entered a well become visible only minutes before the operator must respond, bad hole-cleaning is a situation that may deteriorate gradually over several hours.

Current alarm systems tend to employ simple pattern classification such as threshold values and trend detection, with more sophisticated systems focusing on recognizing the safe events that cause false alarms [13]. In the case of drilling, false alarms are today a major complaint among the users [14]. Attempts at pattern recognition by supervised learning may learn to foresee these common events, but the most severe events are rare in comparison. With few examples, a straight-forward approach taking into account all system parameters and using a large sliding window is then bound to experience the "curse of dimensionality” [15].

3.2 The Hierarchy

To get a grip on the data and underlying processes, one approach is a hierarchical decomposition. In [16] Saputelli et.al introduced the “Field Operations Hierarchy” in Figure 2 as a convenient structuring for the problem of optimizing the production of oil and gas.

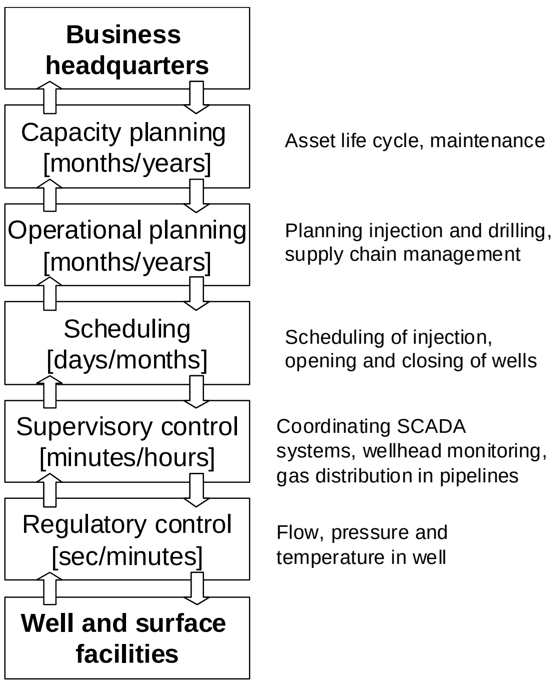

Figure 2: The Field Operations Hierarchy according to Saputelli

The structure will be familiar from other industries. In this figure, information travel upwards and orders are sent downwards. Both scales of time and space increase for higher levels. These levels are a result not just of management structure but of the time-scales of the physical processes involved. For instance, a flow measurement has a time-scale of seconds and may relate to a branch of a single well. The measurement is relayed to the scheduling level which may plan for days ahead taking the gradual wear of equipment into account. Operational planning in turn must plan for the even slower depletion of the whole oil-field.

Orders are subsequently relayed downward e.g. for the closing of valves in the well. This forms a closed loop of supervisory control, where time series fault detection and prediction as well as predictive control becomes important.

Such a hierarchy draws on theory from supervisory control theory, where such nested loops may also be associated with the supervisor's learning process [17, 18].

3.3 The characteristics of different levels

In addition to being a layout for optimization problems, the hierarchy is a useful roadmap for time series prediction. It appears that the demands placed on a time series prediction system depends very much on where in the hierarchy it is implemented. One may for instance notice that the information relayed becomes increasingly symbolic and aggregated as one move upward in the hierarchy. From numerical values that are interpreted higher up as states of the equipment and status reports, on to "net present value" at headquarters. It is telling that we find a symbolically based method like Case Based Reasoning analyzing job reports in the day to month range [19, 20], while typical applications of more numerical methods like neural networks focus on the lower levels [21-23]. In the lower levels it also usually demanded that we restrict ourselves to algorithms that work in real-time systems.

An exception to the symbolic trend is the task of simulating oil and gas reservoirs. This deals with large scales of time and space but mainly numerical data. Prediction of the movement of gas, oil and water in the rock is a computing-intensive problem, made harder by sparse measurements.

Soft computing on time series is here found in two niches. The first is as an aid in history-matching of the model. With many free parameters and much time spent on each run, it is tempting to use soft computing methods to optimize the parameter search. Efforts include evolutionary algorithms [24] and ensemble Kalman filters [25]. This also allows us to use deterministic models while moving towards a probabilistic assessment of subsurface conditions. This probabilistic viewpoint is another trend in the petroleum industry made possible by increased computing power.

The second application sees the time-consuming simulator replaced by a surrogate model, such as a neural network. Trained on input and output from a traditional model, the neural network gives quicker predictions, allowing us to e.g. try out a larger number of different well placements, or explore more of the parameter space. This approach is sometimes referred to as "neuro-simulation" in the literature [26].

Moving down to real-time measurements, a typical issue here is the non-stationarity. Time series from drilling record a system with frequent exogenous inputs, as the drilling operator frequently intervenes to change rates of flow, pipe rotation or type of fluid used. A drilling operation is composed of several different tasks and a parameter value that indicates imminent danger in one situation may be in the normal range in another. The classification of "drilling modes" would therefore feature prominently as a pre-processing step on the way to more sophisticated fault predictions.

The drilling mode classification is also becoming an increasingly pressing issue for symbolic analysis at the higher levels. Much of the system knowledge gathered by methods such as CBR derives from human-made logs and reports of operations. But if such systems are to offer analysis and advice in real-time, they would need real-time reports. A drilling mode classification could correspond to such reports, which shows how applications of hybrid systems may arise naturally in the field operations hierarchy.

Recent efforts at automated classification include a rule-based system by Thonhauser et.al. for the automatic generation of drilling reports [27, 28], but the problem of a reliable *real-time* classification is still an unsolved problem.

4 Combined approaches

The hierarchical approach gives us some leads on overcoming the curse of dimensionality, but not all methods rely on this. For instance, in [29] Lorentzen et.al study an optimization problem where they make a leap directly from choke control to net present value. A common factor in their approach and the previously discussed soft computing methods in reservoir simulation is the combination of soft computing with physical models. Advanced simulators exist for all levels from reservoirs to well drilling [5] and is in a sense an encoding of our knowledge of the system.

It is recognized in system identification and grey-box modelling [30] that "fictious data" is a convenient way to encode expert knowledge, which the simulators readily provide. It is the author's opinion that a combined hard and soft computing approach would be viable not only for the aforementioned optimization problems, but also for fault prediction in time series. However, as mentioned, the physical models do not necessarily reproduce fault signatures; properties of the noise or some complex effects may lead to false alarms.

An approach taken by e.g. Forssell and Lindskog in [31] is to run the best available model alongside measurements and train the AI on their difference or the unexplained "residual". That is, to predict:

$$T_{residual} = T - T_{model\ prediction}$$

We may then re-order the equation to yield an improved prediction:

$$T_{combined\ prediction} = T_{Prediction\ of\ residual} + T_{Model\ \ prediction} \approx T$$

This improved prediction may in turn be used to remove false alarms or increase the sensitivity of established fault detection methods, as implemented by this author in [32]. However, this approach tends to assume that the task of predicting $T_{residual}$ is a simpler or lower-dimensional task than the prediction of T. While often true, counterexamples show that this is not true in general. Other possibilities for injecting prior knowledge from simulations exist, but the author is not aware of well-established methods for the general case.

5 Conclusions

Petroleum exploration and production is an industry that provides researchers with multivariate time-series with challenging "real-world" properties. The time series call for different prediction tasks which seem suited to wildly different schools of prediction systems, while at the same time hinting at a need for a "deeper", perhaps hybrid, system architecture.

Regarding applied research and commercial applications of time series prediction, we find that management now has an open mind towards new methods, under the umbrella of Integrated Operations. However, applications such as real-time fault detection will find that there is a low tolerance of false alarms while time series prediction as part of e.g. production optimization, would have to compete against successful traditional methods. To find acceptance in the industry, and more importantly, to be useful, it is the authors' opinion that time series prediction results must be in a form that can be combined with those from existing physical models. This approach has the potential of yielding better accuracy, stability and generalisation capability than each method alone. It would also be in the spirit of Integrated Operations for us to integrate the experience inherent in time series with the knowledge inherent in physical models.

References

[1] J. Liebowitz, *The handbook of applied expert systems*, Boca Raton, FL: CRC Press, 1998.

[2] M. H. Amara, and B. Martin, "SPE 20419: The Offshore Directional Drilling Advisor: An Expert System for Directional Drilling Optimization," in SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 1990.

[3] N. Hytten, L. Havrevold, and P. Parigot, "SPE 23052: Getting More Out of Drilling Data by Analysis-While-Drilling," in Offshore Europe, 3-6 September 1991, Aberdeen, United Kingdom, 1991.

[4] D. Govia, and K. Carpenter, "SPE 112237-MS: Valuation Models for Intelligent Strategies," in Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2008.

[5] R. Rommetveit, K. S. Bjørkevoll, S. I. Ødegård *et al.*, "Automatic Real-Time Drilling Supervision, Simulation, 3D Visualization and Diagnosis on Ekofisk," in 2008 IADC/SPE Drilling Conference, Orlando, Florida, 2008.

[6] Nygaard, Jahangir, Gravem *et al.*, "SPE 112742-MS: A Step Change in Total System Approach Through Wired Drillpipe Technology," in IADC/SPE Drilling Conference, Orlando, Florida, 2008.

[7] T. S. Olberg, H. Laastad, B. Lesso *et al.*, "SPE 112702-MS: The Utilization of the Massive Amount of Real-Time Data Acquired in Wired Drillpipe Operations," in IADC/SPE Drilling Conference, Orlando, Florida, USA, 2008.

[8] A. Shere, Y. Roberts, and S. Bakkevig, "SPE 112130-MS: Online Production Optimisation on Ekofisk," in Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2008.

[9] X. Wang, J. Lee, B. Thigpen *et al.*, “SPE 111790-MS: Modeling Flow Profile Using Distributed Temperature Sensor (DTS) System,” in Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2008.

[10] M. Hernandez, D. MacNeill, M. Reeves *et al.*, “SPE 113157-MS: High-Speed Wired Drillstring Telemetry Network Delivers Increased Safety, Efficiency, Reliability and Productivity to the Drilling Industry,” in SPE Indian Oil and Gas Technical Conference and Exhibition, Mumbai, India, 2008.

[11] R. Ahmed, and S. Miska, “SPE 112604-MS: Experimental Study and Modeling of Yield Power-Law Fluid Flow in Annuli with Drillpipe Rotation,” in IADC/SPE Drilling Conference, Orlando, Florida, 2008.

[12] M. Khasanov, R. Khabibullin, and V. Krasnov, “SPE 88557-MS: Interactive Visualization of Uncertainty in Well Test Interpretation,” in SPE Asia Pacific Oil and Gas Conference and Exhibition, Perth, Australia, 2004.

[13] D. Hargreaves, S. Jardine, and B. Jeffryes, “SPE 71369 - Early Kick Detection for Deepwater Drilling: New Probabilistic Methods Applied in the Field,” in SPE Annual Technical Conference and Exhibition, New Orleans, 2001.

[14] P. S. A. Norway, *Human factors i bore og brønnoperasjoner - Borernes arbeidssituasjon*, 2007.

[15] M. Verleysen, “Learning high-dimensional data,” *Limitations and Future Trends in Neural Computation, IOS Press*, pp. 141-162, 2003.

[16] L. Saputelli, H. Malki, J. Canelon *et al.*, “SPE 77703-MS, A Critical Overview of Artificial Neural Network Applications in the Context of Continuous Oil Field Optimization,” in SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 2002.

[17] T. B. Sheridan, "Supervisory Control," *Handbook of Human Factors*, G. Salvendy, ed., pp. 1243-1268, 1987.

[18] J. Rasmussen, "Information Processing and Human-Machine Interaction, An Approach to Cognitive Engineering," 1986.

[19] P. Skalle, and A. Aamodt, "Knowledge-Based Decision Support in Oil Well Drilling," *Intelligent Information Processing II*, pp. 443-455, 2005.

[20] Popa, Popa, Malamma *et al.*, “SPE 114229-MS: Case-Based Reasoning Approach for Well Failure Diagnostics and Planning,” in SPE Western Regional and Pacific Section AAPG Joint Meeting, Bakersfield, California, 2008.

[21] R. K. Fruhwirth, G. Thonhauser, and W. Mathis, "SPE 103217, Hybrid simulation using Neural Networks to Predict Drilling Hydraulics in Real Time," 2006.

[22] M. et.al, *Method and apparatus for prediction control in drilling dynamics using neural networks*, USA US 6,732,052 B2,to Baker Hughes Incorporated, 2004.

[23] Dashevskiy, Dubinsky, and Macpherson, “56442-MS: Application of Neural Networks for Predictive Control in Drilling Dynamics,” in SPE Annual Technical Conference and Exhibition, Houston, Texas, 1999.

[24] Selberg, Ludvigsen, Diab *et al.*, “SPE 102349-MS: New Era of History Matching and Probabilistic Forecasting--A Case Study,” in SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 2006.

[25] Lorentzen, Nævdal, Vallès *et al.*, “SPE 96375-MS: Analysis of the Ensemble Kalman Filter for Estimation of Permeability and Porosity in Reservoir Models,” in SPE Annual Technical Conference and Exhibition, Dallas, Texas, 2005.

[26] Hari, Ertekin, and Grader, “SPE 39962-MS: Methods of Neuro-Simulation for Field Development,” in SPE Rocky Mountain Regional/Low-Permeability Reservoirs Symposium, Denver, Colorado, 1998.

[27] G. Thonhauser, “Using Real-Time Data for Automated Drilling Performance Analysis,” *Oil Gas European Magazine*, no. 4/2004, pp. 170-173, 2004.

[28] G. Thonhauser, and W. Mathis, "SPE 103211, Automated Reporting Using Rig Sensor Data Enables Superior Drilling Project Management," 2006.

[29] R. J. Lorentzen, A. M. Berg, G. Nævdal *et al.*, “SPE 99690-MS: A New Approach for Dynamic Optimization of Waterflooding Problems,” in Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2006.

[30] M. Kárný, A. Halousková, and P. Nedoma, “Recursive approximation by ARX model: a tool for grey box modelling,” *International journal of adaptive control and signal processing*, vol. 9, no. 6, pp. 525-546, 1995.

[31] U. Forssell, and P. Lindskog, “Combining semi-physical and neural network modeling: an example of its usefulness,” in 11th IFAC Symposium on System Identification (SYSID'97), 1997.

[32] R. Nybø, K. S. Bjørkevoll, and R. Rommetveit, “SPE 112212-MS, Spotting a False Alarm—Integrating Experience and Real-Time Analysis With Artificial Intelligence,” in Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2008.

Paper IV

Is not included due to copyright