

TESTING FOR RELIABILITY
AN APPLIED APPROACH

By
C.J.L.Rothkrantz

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
DELFT UNIVERSITY OF TECHNOLOGY
OCTOBER 28, 2005
IN COOPERATION WITH
TNO Building and Construction Research

© Copyright by C.J.L.Rothkrantz, 2005

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Operations Research and Risk Analysis for acceptance a thesis entitled “**Testing for Reliability An Applied Approach** ” by **C.J.L.Rothkrantz** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: October 28, 2005

Supervisors:

prof.dr. R.M.Cooke

dr. M.S. de Wit

Readers:

dr. D. Kurowica

dr. T.A. Mazzuchi

DELFT UNIVERSITY OF TECHNOLOGY

Date: **October 28, 2005**

Author: **C.J.L.Rothkrantz**
Title: **Testing for Reliability An Applied Approach**
Department: **Electrical Engineering, Mathematics and Computer Science**
Degree: **M.Sc.**

Permission is herewith granted to Delft University of Technology and TNO Building and Construction Research to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

To Claire.

Table of Contents

Table of Contents	ix
List of Tables	xi
List of Figures	xii
Abstract	xv
Acknowledgements	xvii
Introduction	xix
1 Bayesian Theory	1
1.1 Subjective Probability	1
1.2 Bayes' Theorem	2
2 The Timber Case	5
3 The Bayesian Method	9
3.1 Definition of the Method	9
3.2 Application of the Method to an Example	9
3.3 Improper Priors	12
4 Different Approaches of the Bayesian method	15
4.1 Analytical Approach	15
4.2 Direct Numerical Integration Approach	18
4.3 BBN Approach	18
4.4 Bayes Linear Approach	20
5 Application of the Different Approaches to the Timber Case	25
5.1 Application of the Analytical Approach	25
5.2 Application of the Direct Numerical Integration Approach	31
5.3 Application of the BBN Approach	34
5.4 Application of the Bayes Linear Approach	36
6 Decision Problem	39
6.1 Introduction to the Decision Problem	39
6.2 Savage's Decision Theory	39

7	Approaches to the Decision Problem	43
7.1	Considerations Decision Problem	43
7.2	Defining the Decision Problem	45
7.3	Approaches Solving Decision Problem	45
7.4	Application to Timber Case	48
7.4.1	Considerations Timber Case	48
7.4.2	Application Decision Tree	49
7.4.3	Application Fixed Size Test Plan	51
7.4.4	Application VOI	56
8	General Method for Testing	59
8.1	The General Method	59
8.2	Humidity Case	60
9	Conclusions, Recommendations and Further Research	69
9.1	Conclusions	69
9.2	Recommendations	70
9.3	Further Research	70
A	Appendices to the Text	71
A.1	Proof Multiple Destructive Observations	71
A.2	Calculations and Proofs to Propositions Concerning Comparison Bayes Linear	72
A.3	Proof Characteristic Value of the Timber	74
A.4	Proofs Bayes Decision Rules	75
A.5	Direct Numerical Integration in Humidity Case	77
B	Controlling HUGIN	79
B.1	Introduction	79
B.2	Installation	80
B.3	Visual Basic for Excel	80
B.4	MATLAB	83
B.5	Complete Examples MATLAB	88
C	Codes	91
D	Notation	117
	Bibliography	121
	Index	123

List of Tables

5.1	Test results for Camaru timber ($[N/mm^2]$)	25
5.2	Parameters prior distributions	25
5.3	Observed destructive strengths	30
5.4	First two moments of R determined with Bayes linear and direct numerical integration	36
6.1	Expected profit of the jazz concert	40
6.2	Information on the weather-test	41
7.1	Parameters for the normal distributions used for all nodes in the decision tree	50
7.2	Input for decision problem	54
7.3	Classification of characteristic strength (NEN6760)	54
8.1	Input for decision problem	61
8.2	Conditional probability table of node error1 in HUGIN	63
8.3	Discretisation for T_{Dry} in HUGIN	65
B.1	Possible types of nodes	85

List of Figures

1	A graphical representation of the Bayesian method	xvi
1.1	Reverend Thomas Bayes (1702-1761)	2
3.1	A graphical representation of the Bayesian method	10
3.2	A plot of the Bayesian update	11
3.3	A plot of the prior and predictive density for R	12
3.4	A visualization of the applied Bayesian method	13
4.1	General belief net representation for the timber case	19
4.2	Prior and posterior distribution in HUGIN	20
4.3	A plot of the posterior distribution of M via Bayes linear and direct numerical integration in the simplified timber case	23
5.1	A plot of the prior information modelled by a normal inverted gamma density	31
5.2	A plot of the prior and predictive distribution	32
5.3	Posterior distributions for M, Σ using an informative and vague prior with the Camaru data-set	33
5.4	The predictive distribution of R using vague and an informative prior with the Camaru data-set	34
5.5	The graphical model from HUGIN used in the timber case	35
5.6	A plot of the posterior distribution derived by direct numerical integration and HUGIN	36
6.1	The optimal decision for the location of the concert displayed in a decision tree, where the (maximal)expected profit is displayed in the squared brackets	41
6.2	The decision problem to perform a test to decide the location of the concert displayed in a decision tree	42
7.1	A plot of the expected utility and costs against the number of performed tests	44

7.2	The decision tree in the timber case for at most two test and a discretisation of three intervals	50
7.3	A continuous version of the decision tree in the simplified timber case evaluating at most two tests	51
7.4	The decision loss against the number of test in the timber case	52
7.5	A plot of the expected loss against the number of tests	56
8.1	Schematized view of a wall under investigation	60
8.2	Building the graphical representation of the Bayesian method	66
8.3	The graphical model for the humidity case	67
8.4	Results of the fixed size test plan approach in the humidity case	68
B.1	The Example network in HUGIN created via VB	80
B.2	A plot of an interval with the index used in HUGIN	82
B.3	The example network in HUGIN	88

Abstract

Increasing performance demands have led to the need for better modelling of the behavior of products and constructions. These models are needed to design and construct these constructions or products, as well as to verify that they meet the demands (or will in the future). The models that are used are subject to uncertainties. As a result of these uncertainties products and constructions are over-designed (safety margins are used) or are very conservative modelled. This increases the cost of these products and constructions. Therefore it is investigated if performing tests can lower these costs, by reducing the uncertainties. Note that tests are not for free. Therefore the expected savings in costs of the product or construction by reducing the uncertainties have to be weighed against the cost of performing the tests.

The following method has been formulated to handle this problem.

General Method for Testing

Problem definition

1. Define testing objectives and decision problem,
2. Create models,
3. Define primary variables,

Interpretation of test results

4. Apply Bayesian method,
5. Validation of Bayesian method and models,

Test plan

6. Define optimal test plan,

Evaluation

7. Interpret results.

In this method there are two main steps, respectively ‘Apply Bayesian method’ and ‘Define optimal test plan’. These two steps are discussed separately in much detail, and for both, different approaches have been derived and reviewed by application to two cases, respectively the timber case and the humidity case.

The Bayesian method makes it possible to interpret test results and reduce the uncertainties. A visualization of this method is given in figure 1. To implement the Bayesian method the following four approaches have been discussed;

The Bayesian Method

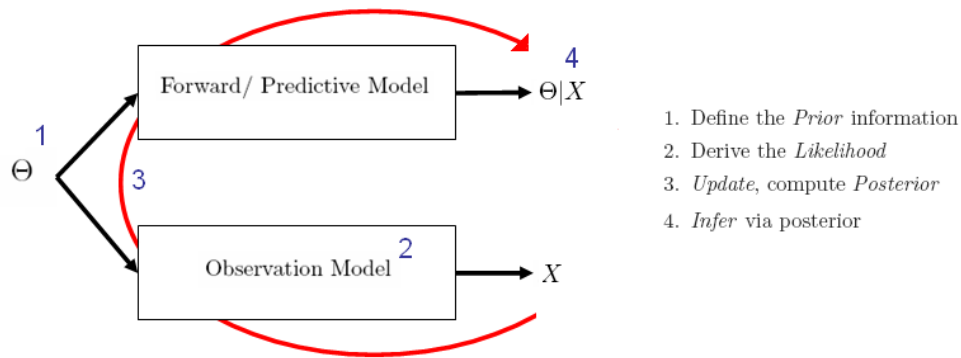


Figure 1: A graphical representation of the Bayesian method

- Analytical approach (conjugate analysis),
- Direct numerical integration approach,
- Bayesian belief net approach,
- Bayes linear approach.

The Bayesian belief net (BBN) approach was chosen to use for modelling a general case.

To obtain the optimal number of tests a decision problem has to be defined. This decision problem can be solved using Savage's decision theory, which needs a utility function to measure the effects of tests. In order to help defining this utility function two possible approaches have been proposed and applied to two cases, respectively the producers and designers approach. Both use a different perspective to define the utility function, where the one uses sales revenues and production costs the other uses maintenance/repair costs and costs of failure to solve the problem. For calculating the solution to the decision problem three approaches have been investigated:

- Decision tree approach,
- Fixed size test plan approach,
- Value of information approach.

One of the main deliverables of this thesis is the numerical tool that was created in MATLAB to solve such decision problem. This program uses the BBN-approach and the fixed size test plan to solve the decision problem. In order to write this code HUGIN, a program that can handle BBNs, has been linked to MATLAB using the HUGIN API. A manual explaining how this is done is added in the Appendix.

Acknowledgements

These acknowledgements and thanks recognize the primary contributors that have helped me during my thesis research. Therefore I would like to thank my supervisors from Delft University of Technology, R.M. Cooke, D. Kurowica, and T.A. Mazzuchi for all their help and comments. I think the ever so occasional conversations with them on my thesis, is how they have helped me the most. In my opinion the Risk and Environmental Modelling study-program is very unique and outstanding!

Most important I would like to thank my daily supervisor from TNO Building and Construction Research M.S. de Wit. This not only for his supervision, comments, and remarks, but also for his showed understanding, in for me personally, difficult times during my thesis research. Therefore I would also like to thank TNO Building and Construction Research for giving me the opportunity to write my thesis within their organization. I found it a very stimulating research environment and it has proven its high level of quality. Not only on an intellectual basis but also socially. Moreover I will never forget the ‘C4-breaks’ with Sten, Wouter, Wouter, Wietske, Alfons, and Wim. They have contributed to this thesis far more then I could have ever expected.

Also I would like to thank my fellow students who have helped me reviewing previous version and supplying me with many suggestions and comments. To some of them also thanks for the occasional help with Latex to typeset this report, does it not look good? In particular I mention Martijn and Sander who have helped me the most.

Finally I would like to thank my parents, brothers, (soon to be) parents-in-law and wife for supporting me during my thesis research and study. They have helped me not only by giving me support but also helped me concerning the contents of this thesis. Without their help I would not have been able to finish my thesis and conclude my study at Delft University of Technology.

Introduction

Modern day products and constructions are subject to increasing performance demands. More precisely a construction is expected not to fail during its intended lifetime. Therefore during design models are used to show whether a construction's performance can meet the issued demands. The models used are subject to uncertainties which can vary from correctness of the model to uncertainty over input parameters. This introduces uncertainty over the performance. In order to fulfill the performance demands, safety margins are used in the design of the constructions or products. This makes the construction or product more costly. If these uncertainties can be reduced the design can be done more efficiently, and still fulfill the performance demands.

In many cases empirical data obtained from tests can be used to reduce these uncertainties. This requires a method that incorporates the empirical data received from observations with the (mathematical) models that are used for modelling the performance of the construction or product. This thesis shows our contributions towards identifying that methodology and extending it.

Objectives

Therefore the objectives of this thesis are stated as to identify and extend the methodology that can incorporate empirical data in the reliability models. This methodology will be extended such that it can also address the decision-dilemma, mentioned above, to either perform a number of tests or not. This with the aim to increase, better determine, the reliability of a product or construction by reducing the uncertainties via testing. Subsequently, a secondary objective is to create a (generic, numeric) tool which helps to apply this methodology.

Outline of the Report

This report can be divided into three parts. The first part consists of Chapters one to five. This part addresses the issue of how to incorporate the observations into the mathematical models when observations have been obtained (updating). This means that the observations are considered as given. It explains the Bayesian method and discusses four different approaches, subsequently they are applied to a specific case. In the second part consisting of Chapter six and seven, a decision problem is defined to determine the testing procedure. This means that the observations are seen as random variables. The third part, Chapter eight, will describe the entire derived methodology of testing for reliability. It does so by an application of the most suitable approaches derived in the first two parts and combines them. The thesis ends with a conclusion in Chapter nine.

The appendix consists of some sections containing additional mathematical details to the text in the previous chapters, such as proofs and (long) calculations. The appendix also contains the codes

implemented in MATLAB (and DPL). Further a list of used notation is added, as is a bibliography and index.

Previous work

TNO Building and Construction Research has often investigated and/or validated the reliability of existing constructions using measurements. This has resulted in many case reports, in which an individual method is applied to solve the problem. These reports¹ gave insight to the problems under investigation and the methods(methodology) already used at TNO Building and Construction Research, which will be identified as the '*Bayesian method*'. There were also some documents that addressed the issue more general [5], [18], these documents mostly used models based on the normal distribution. These documents showed me that there is a lot of knowledge on specific areas of application which also explains the need to combine these individual methods into this general methodology.

During the research for this thesis at TNO Building and Construction Research, a lot of work was done on a specific case discussed in this report as the timber case. This case has proven a good application of the theory discussed in this thesis. Where needed references are given to the previous work done at TNO Building and Construction Research and to the studied literature.

¹See for instance [22], [8], and [9].

Chapter 1

Bayesian Theory

This chapter will set forth the theory needed to deal with the issue of interpreting test results¹ in concordance to the information that was at hand before testing. This is needed to define an approach that will be used to make inferences on the reliability of a structure. The needed theory is called Bayesian theory, which can not be discussed without the notion of Subjective Probability. Therefore this topic will be handled first.

1.1 Subjective Probability

Bayesian Theory is based on the notion of *subjective probability*. This is different from the classical and frequentist perception of probability. In the classical view the probability of an event is defined (by Laplace) as the number of equally likely outcomes obtained from repeated trials that lead to the event, divided by the total number of equally likely outcomes. Underlying this view of probability is the notion of symmetry. Physical symmetry implies equal probability and if it is impossible to tell which outcome is more likely they should be assigned equal probability. This view on probability posed some problems resulting in defining the *frequentist interpretation*. This interpretation defines the probability of an event as the frequency of occurrence of that event (in an infinite sequence of hypothetical trials).

Subjective probability does not necessarily do so. The probability of A , denoted as $P(A)$, reflects a persons degree of belief of occurrence of A measured on a scale $[0, 1]$. The assignment $P(A) = 1$ means A occurs, $P(A) = 0$ means A does not occur. Furthermore all the rules of probability still apply. Notice that A need not be an event or a result of (an infinite sequence of) repeated trials. Therefore the notion of subjective probability is richer than the classical or frequentist. Using the subjective probability interpretation can lead to the same probability as when using the other views if it is believed that A can be observed by trials. The notion of subjective probability was developed into a full axiomatic theory by Leonard J. Savage [14].

¹Throughout this report the terms observations, measurements, and test results will be used interchangeably.

1.2 Bayes' Theorem

'Bayesian methods' is a general name for methods that use the (implications of the) theorem defined by reverend *Thomas Bayes* . Bayes lived from 1702 to 1761 and formulated his theorem to answer the problem of inverse probability. Many applications of this theorem have only recently been found. For more historical details on Bayesian Methods and Thomas Bayes, we refer to [1], [2].



Figure 1.1: Reverend Thomas Bayes (1702-1761)

The basic form of *Bayes' theorem* (for events) is given by the theorem below.

Theorem 1.2.1 (Bayes' Theorem for single events). *Let A and B be two events with $P(A) > 0$, then*

$$P(B|A) = \frac{P(A|B)}{P(A)}P(B). \quad (1.2.1)$$

Proof. This theorem is a direct result of the following identity in probability theory

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A). \quad (1.2.2)$$

□

From the above proof it can be seen that Bayes' theorem is a direct result of the definition of conditional probability. Many people would think that this therefore is a trivial theorem. Nevertheless, the interpretation that can be given to Equation (1.2.1) is not trivial (and this is what Bayes did!). Bayes interpreted the left hand side (lhs.) of this equation as the revised probability of occurrence of event B after observing event A . He called this conditional probability $P(B|A)$ the posterior (probability) of B . This makes it possible to revise probabilities with the help of observations. The

probability $P(B)$ on the right hand side (rhs.) of this equation is called the prior (probability), representing the probability of B before the observation is performed and $P(A|B)$ is called the *likelihood (function)*² reflected as the probability of the occurrence of event A (the observation) under the hypotheses (B has occurred). The term $P(A)$ is just considered as a scaling factor. Thus by using Equation (1.2.1) it is possible to make inference on the event B by observing event A. Notice that if A and B are independent no inference can be made.

However most inference problems do not concern discrete events. Therefore Bayes' theorem is usually stated in terms of density functions, called Bayes' rule. When interested in making inference on the parameter θ by observing X , the data, both considered as random variables, *Bayes rule* can be formulated as follows.

Theorem 1.2.2 (Bayes' rule). *Let $f_{\Theta}(\theta), f_X(x)$ be the density function of Θ, X respectively, then*

$$f_{\Theta|X}(\theta; x) = \frac{f_{X|\Theta}(x; \theta)f_{\Theta}(\theta)}{f_X(x)}. \quad (1.2.3)$$

Remark 1.2.1. Sometimes Equation (1.2.3) is also denoted as

$$f_{\Theta|X}(\theta; x) = C_1 f_{X|\Theta}(x; \theta) f_{\Theta}(\theta), \quad (1.2.4)$$

where $C_1 = f_X(x) = \int f_{X|\Theta}(x; \theta) f_{\Theta}(\theta) d\theta$ is a normalizing constant.

An often used theorem in Bayesian methods is the *Law of Total Probability(LTP)*. This theorem will be stated here for completeness. The application of LTP is first performed in Chapter 3.

Theorem 1.2.3 (The Law of Total Probability). *Let $A_i, i = 1 \dots n$ be a sequence of countable mutually exclusive events³ and $\bigcup_{i=1}^n A_i \supseteq B$ and let B also be an event. Then*

$$P(B) = \sum_{i=1}^n P(B|A_i)P(A_i). \quad (1.2.5)$$

Proof. Because $\bigcup_{i=1}^n A_i \supseteq B$ and the definition of conditional probability

$$P(B) = \sum_{i=1}^n P(A_i \cap B). \quad (1.2.6)$$

Using the definition of conditional probability on the term in the sum gives the stated theorem. This proof has been taken from [21]. □

Remark 1.2.2. The LTP can also be rephrased in terms of density functions. Then, LTP reads

$$f_B(b) = \int f_{B|A}(b; a) f_A(a) da. \quad (1.2.7)$$

²In a sense, likelihood works backwards from probability: given B , we use the conditional probability $P(A|B)$ to reason about A , and, given A , we use the likelihood function $P(A|B)$ to reason about B . *This text was taken from [3].*

³Two events are mutually exclusive when they have no outcomes in common, e.g., $A_i \cap A_j = \emptyset$ for $i \neq j$.

Chapter 2

The Timber Case

This chapter describes the (general) *timber case*. The aim in this case is to classify the strength(R) of a population of beams of a certain kind of timber. The following model is used to describe the strength of such a beam from this population. The strength (resistance) of a randomly selected beam from this population is assumed to follow a normal distribution¹, where the two parameters of the normal distribution can be interpreted as follows; the mean as the average strength of all the beams from that kind of timber and the standard deviation as the spread in the strengths among all the beams. This spread is the natural variation in the strength of the timber. This natural variation arises from the natural growth process of the timber. This model is denoted as

$$R|\mu, \sigma \sim \mathcal{N}(\mu, \sigma). \quad (2.0.1)$$

The parameters of this normal distribution μ and σ are both unknown and are modelled as stochastic variables themselves² to incorporate uncertainty into the model. This is implemented in the notation by denoting them with capital letters. This as capital letters denote stochastic variables and lowercase letters denote realizations or integration variables of the stochastic variables. The model is denoted as

$$R|(M, \Sigma) \sim \mathcal{N}(M, \Sigma). \quad (2.0.2)$$

Next we want to use the prior information (if available) and tests/measurements to determine the strength by determining the distributions over the parameters. Different (Bayesian) approaches can be implemented. For instance different distributions can be chosen to model the distribution over the parameters, different data and computational methods can be used. Some of these different settings will be discussed in the following chapters.

A simplification that will also be used as an example in the following chapters is called the *simplified timber case*. This is actually the same setting as the (general) timber case with the difference that not both parameters of the (normal) distribution over the strength of the timber(R) are modelled as stochastic variables, only the mean(M) is modelled as a stochastic variable,

$$R^{simp}|(M, \sigma) \sim \mathcal{N}(M, \sigma). \quad (2.0.3)$$

The data that is available in the timber case can be categorized in two groups. The first, *destructive data*(D) is data derived from destructively strength testing of a random selected beam

¹The verification of this model has been done in [19].

²Which on its own relegates to the Bayesian approach. In classical/frequentative approaches these parameters can only be a fixed value.

from the population. Note that in this case each beam tested can not be used after testing. If more than one measurement is available it consists of tests done on different beams (random) selected from the population of beams. The destructive data, can be represented as random (independent) realizations or samples from the distribution of the strength of the timber,

$$D_i = r_i. \quad (2.0.4)$$

This causes the likelihood to be a normal distribution with parameters M , and Σ , e.g.,

$$\begin{aligned} L_{D|M,\Sigma}(d; \mu, \sigma) &= f_{R|M,\Sigma}(d; \mu, \sigma) \\ &= \frac{1}{\sigma} \varphi\left(\frac{d - \mu}{\sigma}\right), \end{aligned} \quad (2.0.5)$$

in which the Greek letter, φ , denotes the standard normal distribution. Due to conditional independence given the parameters of the distribution, the multi-dimensional likelihood which needs to be used when multiple destructive measurements are used is the product of normal densities with parameters M , and Σ .

$$\begin{aligned} L_{\underline{D}|M,\Sigma}(\underline{d}; \mu, \sigma) &= \prod_{i=1}^n f_{R|M,\Sigma}(d_i; \mu, \sigma) \\ &= \frac{1}{\sigma^n} \prod_{i=1}^n \varphi\left(\frac{d_i - \mu}{\sigma}\right). \end{aligned} \quad (2.0.6)$$

Censored data derived from destructive measurements can also be used in this setting, it was discussed in [18]. This type of data will not be used in this thesis.

The second group of data is that of the *non-destructive data*. The data in this group is derived from non-destructive testing on a beam of timber. Notice that this implies that a beam can be tested and can also be used after testing, which is an important difference compared to destructive testing. To model this type of data received from non-destructive tests two different models are generally used. Unlike destructive testing non-destructive testing is subjected to a measurement error. The following two models are generally used to model this error:

- Additive Error Model,
- Multiplicative Error Model.

In the first error model, the *additive model* is used to model the error in the non-destructive measurements. The following equation can be formulated for the non-destructive Data(Υ) obtained from testing a randomly selected beam,

$$\Upsilon = R + \varepsilon, \quad (2.0.7)$$

where the 'error-term'(ε) is assumed to have mean 0 and follow a normal distribution, thus $\mathcal{N}(0, \sigma_\varepsilon)$. When the second model is used the equation reads

$$\Upsilon = R * \varepsilon, \quad (2.0.8)$$

where the 'error-term' is assumed to have mean 1 and follow a normal distribution, thus $\mathcal{N}(1, \sigma_\varepsilon)$. The likelihood of the non-destructive data is a bit different then for destructive data, to derive it we need application of the 'law of total probability'³. This is discussed in Chapter 5 where the timber case is used to implement the Bayesian method.

³This theorem was given in Chapter 1.

The two models for the error, Equations (2.0.7) and (2.0.8) falsely look in formula more or less the same. However, this seemingly small difference has a large influence on the complexity of the calculations needed to be performed when doing a Bayesian update as will be shown in Chapter 5. From a modelling perspective the two models are also very different as the source underlying the error in both cases differs.

The case specifics defining the decision problem are given in Chapter 7, after the decision problem has been defined.

Chapter 3

The Bayesian Method

In this chapter the Bayesian method is formulated and will be illustrated in a simple example by application to the simplified timber case (see Chapter 2).

3.1 Definition of the Method

In its most general form the Bayesian method is formulated as follows, also see [4].

Definition 3.1.1 (The Bayesian Method). The *Bayesian method* (for inference) consists of four steps, namely

1. Define the *Prior* information,
2. Derive the *Likelihood*,
3. *Update*, compute *Posterior*,
4. *Infer* via posterior .

This procedure is graphically represented in Figure 3.1. In this figure all steps are denoted by blue numbers. Let us start to apply the Bayesian method to the simplified timber case to clarify all the steps. While doing so we propose to use the graphical representation of the method depicted in Figure 3.1, as it will help to structure the application.

3.2 Application of the Method to an Example

The Bayesian method was formulated for inference on the parameter Θ , in the simplified timber case this is the mean, M . Therefore in Figure 3.1 read M instead of Θ and as destructive data(D) is going to be used read D instead of X .

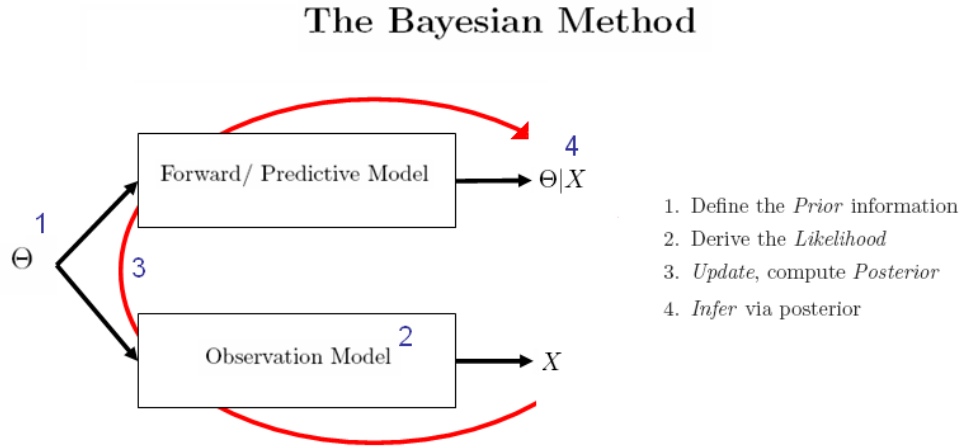


Figure 3.1: A graphical representation of the Bayesian method

The first step in the Bayesian method is defining the *prior information*. In the simplified timber case a-priori M is assumed to follow a normal distribution with known parameters. This is mathematically formulated as

$$M^{prior} \sim \mathcal{N}(\mu_M, \sigma_M). \quad (3.2.1)$$

In Figure 3.2 this prior density is visualized. Notice that in this case a clear interpretation can be given to the prior information. The mean of the prior distribution (μ_M) can be thought of as the best estimate of the mean strength of all the beams. The variance of the prior distribution (σ_M^2) can be thought as a measure of how (un-)certain this value is. Often in applications the prior information is vague or not available. At the end of this chapter a subsection is devoted to the use of certain (improper) priors to model these situations.

The second step in the Bayesian method is to derive the *likelihood*. To derive the likelihood a model has to be available for the data, this as the likelihood represents the plausibility of the hypotheses (the parameter M) represented in the data. In this setting the data is obtained as a sample from R . Therefore the likelihood can easily be derived as

$$L(M) = f_{D|M}(d; \mu) = \frac{1}{\sigma} \varphi \left(\frac{d - \mu}{\sigma} \right). \quad (3.2.2)$$

Although the likelihood function is a conditional probability density function of the data given the hypothesis (parameter M), it has, in Bayes rule, to be handled as a function of μ .

Step three in the Bayesian method computing the posterior density function of M is also called performing the *update*. By definition, Bayes Rule for density functions (Theorem (1.2.2)) states

$$f_{M|D}(\mu; d) = \frac{f_{D|M}(d; \mu) f_M(\mu)}{f_D(d)}, \quad (3.2.3)$$

which by substitution of the functions derived in the previous steps gives the following posterior density, i.e.,

$$\begin{aligned}
f_M^{post}(\mu) &= f_{M|D}(\mu; d) = C_1 f_{D|M}(d; \mu) f_M(\mu) \\
&= C_1 e^{-\frac{1}{2\sigma^2}(d-\mu)^2} e^{-\frac{1}{2\sigma_\mu^2}(\mu-\mu_\mu)^2} \\
&= C_2 e^{-\frac{\sigma_\mu^2 + \sigma^2}{2\sigma^2\sigma_\mu^2} \left(\mu - \frac{\sigma_\mu^2 d + \sigma^2 \mu_\mu}{\sigma_\mu^2 + \sigma^2} \right)^2},
\end{aligned} \tag{3.2.4}$$

where C_1, C_2 are appropriate constants such that the density function is normalized, e.g. integrates out to one. The following result is obtained from the calculation given above in Equation (3.2.4),

$$M^{post} \sim \mathcal{N} \left(\frac{\sigma_\mu^2 d + \sigma^2 \mu_\mu}{\sigma_\mu^2 + \sigma^2}, \sqrt{\frac{\sigma^2 \sigma_\mu^2}{\sigma_\mu^2 + \sigma^2}} \right). \tag{3.2.5}$$

As the posterior for M is derived, that is to say the updating is complete, inference can be made, step 4 of the method. All the available information of the data, D , on the mean, M , is reflected in the posterior distribution over M , and inference can be made by comparing the posterior with the prior. Figure 3.2 shows graphically the update using actual values. The data that was used for the update coincided with the mean of the prior distribution, as a result the posterior distribution has the same mean and a smaller variance. This can be interpreted as the reduction of the uncertainty over M caused by the use of the data(measurement).

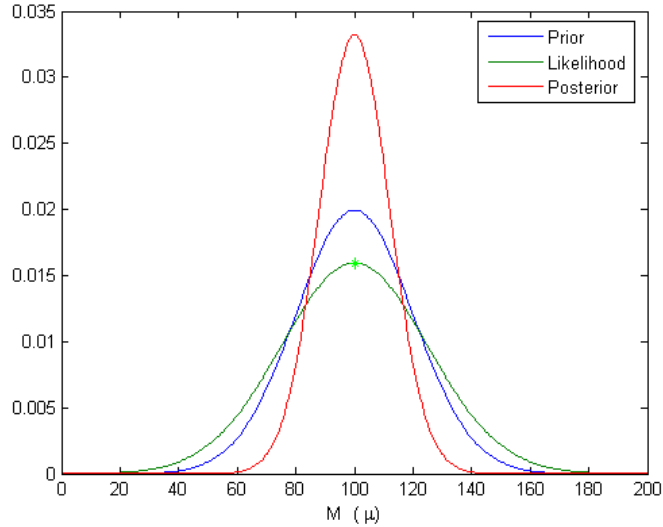


Figure 3.2: A plot of the Bayesian update

In the simplified timber case the primary goal is not to make inference on (the distribution of) the mean(M), it is to make inference on the strength of the timber(R). This is then done via an extra stage in the inference called the predictive model. The predictive model in this case consists of an application of LTP. Recall that the following model for R was assumed.

$$R|M = \mathcal{N}(M, \sigma). \tag{3.2.6}$$

Then, the prior (predictive) distribution for R is given via LTP as

$$f_R(r) = \int f_{R|M}(r; \mu) f_M^{prior}(\mu) d\mu, \quad (3.2.7)$$

which implies a prior distribution for R as $\mathcal{N}\left(\mu_\mu, \sqrt{\sigma_\mu^2 + \sigma^2}\right)$. Using this result the posterior distribution for R (often called the predictive distribution) is found by applying the update to M and then again applying LTP, thus

$$f_{R|D}(r; d) = \int f_{R|M}(r; \mu) f_{M|D}(\mu; d) d\mu. \quad (3.2.8)$$

Resulting in a posterior distribution for R as $\mathcal{N}\left(\frac{\sigma_\mu^2 d + \sigma^2 \mu_\mu}{\sigma_\mu^2 + \sigma^2}, \sqrt{\left(\frac{\sigma^2 \sigma_\mu^2}{\sigma_\mu^2 + \sigma^2}\right)^2 + \sigma^2}\right)$. Now inference can be made on the variable R , for visualization purposes a plot is given displaying the prior and predictive distribution of R in Figure 3.3. Note that the uncertainty is reduced (as can be seen by the smaller

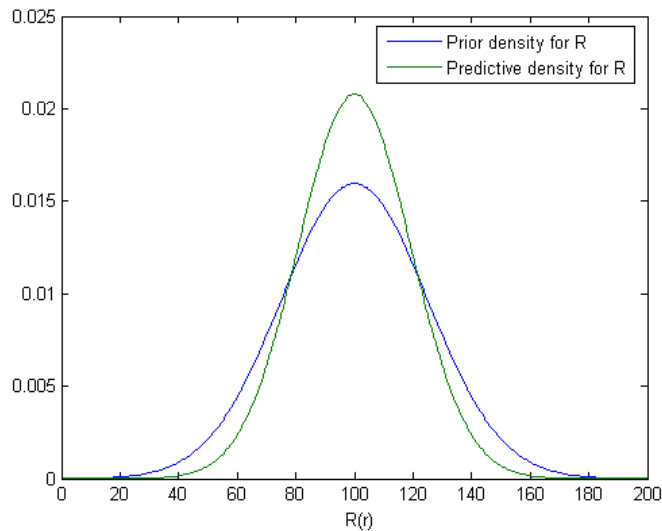


Figure 3.3: A plot of the prior and predictive density for R

variance). Also note that in order to make this inference LTP needed to be applied, applying LTP implies performing an (extra) integration. This can induce a large computational complexity.

This completes the application of the Bayesian method to the simplified timber case. It was advocated to use Figure 3.1 to structure the method. When entering all steps in this figure it should look like Figure 3.4, showing the outline of the applied Bayesian method.

3.3 Improper Priors

This final section of this chapter gives some additional information on a special case of implementing the first step of the Bayesian method, ‘defining the prior’ called using *improper priors*. Improper

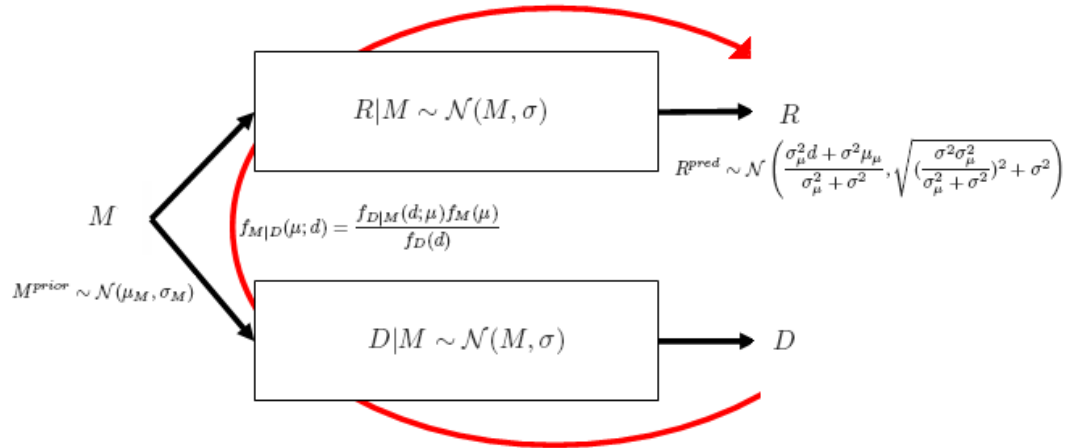


Figure 3.4: A visualization of the applied Bayesian method

priors are discussed now, as throughout the remainder of the thesis occasionally improper priors will be applied.

In a lot of practical situations no specific prior information is at hand, which is then often expressed in ‘wide’ distributions for the prior information. With ‘wide’ distributions is meant distributions with a large variance. As these priors do not hold much information, they are called vague or non-informative priors.

Another approach to model prior information or the lack of, is the use of improper priors. These improper priors are called improper as they do not represent a density function, i.e. the integral over the entire support is not equal to one. An important condition in the use of improper priors is that the posterior must become proper! Often improper priors are used to model limiting cases of proper priors with vague prior information. In Chapter 5 improper priors will be used to model vague prior information. The effect of doing so will also be discussed. A warning must be made on the use of improper priors, the user has to verify that the assumptions on using them can be justified and the results available from the use of them must always be carefully checked.

Chapter 4

Different Approaches of the Bayesian method

The previous chapter defined the Bayesian method and explained it by applying it to an example. In fact a specific approach to apply the Bayesian method, called conjugate analysis, was used. In this section four different approaches to implement and perform the Bayesian method will be discussed. These different approaches can vary in modelling aspect or in the computational methods and/or software that are used. These four specific approaches discussed do by no means form an exhaustive list of all possible approaches. Other approaches are discussed in [4], [21].

The approach best suited to perform the Bayesian method is mostly determined by the problem itself and how it is modelled as this determines the likelihood and the prior distributions. The major factor determining which approach is best used, is the consideration how many times LTP needs to be applied. As in each application of the LTP an integration needs to be performed. This integration influences the ability to perform the approach analytically, or if done numerically, the computation time. This will become apparent when discussing the different approaches.

4.1 Analytical Approach

Performing the multiplications and integrations obtained/needed in the Bayesian method analytically is the foundation for the approach discussed in this section. The integrations can be needed in order to determine the constant in Bayes Rule or for application of the LTP. In some cases the integral can be calculated formally and an analytical solution is found. In that case the update is known exactly and the approach used to perform the update is called analytical. Already an example of this was given in the previous chapter.

Conjugate Approach

A special case of the analytical approach is the *conjugate approach*. In this approach a conjugate analysis is performed, this means that the family of distribution to which the posterior belongs is the same as the prior. When applying a conjugate analysis the update procedure becomes easier and computational less complex, as updating the distribution changes to updating the parameters.

Note that the possibility of performing a conjugate analysis is determined by the likelihood and the prior. In many cases the prior can be chosen but the likelihood can not.

Some families of distributions that have conjugate properties are the exponential, normal, normal-inverted-gamma and the inverse-Wishart distributions. The middle two will be applied throughout this thesis. This subsection will define these two models and display their general use, also some advantages of using these models are discussed. The conjugate properties of the normal distributions (\mathcal{N} -model) were displayed by Equation (3.2.5). Another conjugate model that can be used is the normal-inverted-gamma model (\mathcal{NIG}). This model will now be defined/derived in this section, and the conjugate properties will be shown. It will be applied to the timber case in the next chapter. The following definitions are stated here in order to define the model.

Definition 4.1.1 (The Gamma Distribution). Let X be a *gamma distributed* random variable with shape parameter α and scale parameter β ($X \sim \mathcal{G}(\alpha, \beta)$). The density function of X is given by

$$f_X(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \beta^{-\alpha} e^{-\frac{x}{\beta}}. \quad (4.1.1)$$

The first two moments are $\mathbb{E}(X) = \alpha\beta$, $\text{Var}(X) = \alpha\beta^2$.

Definition 4.1.2 (The Inverse-Gamma Distribution). Let Y be an *inverse gamma* distributed random variable with parameters α and β ¹ ($Y \sim \mathcal{IG}(\alpha, \beta)$). The density function for Y is given by

$$f_Y(y) = \frac{1}{\Gamma(\alpha)} y^{-\alpha-1} \beta^{-\alpha} e^{-\frac{\beta}{y}}. \quad (4.1.2)$$

The first two moments are $\mathbb{E}(Y) = \frac{1}{(\alpha-1)\beta}$, $\text{Var}(Y) = \frac{1}{(\alpha-1)^2(\alpha-2)\beta^2}$.

Definition 4.1.3 (The Normal Inverted Gamma Model). Let $H \sim \mathcal{IG}(\alpha, \beta)$ and $[M|H] \sim \mathcal{N}(m, \sqrt{\eta\Lambda})$ then $[M, H]$ follows a *normal inverted gamma distribution* ($\mathcal{NIG}(\alpha, \beta, m, \Lambda)$) which has the following density function

$$f_{M,H}(\mu, \eta) = \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi} \Gamma(\alpha) \Lambda^{\frac{1}{2}}} e^{-\frac{1}{2} \frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}}. \quad (4.1.3)$$

The definition clearly shows where the name of the model comes from, its density is a product of a normal and an inverted gamma density. The conjugate properties of this model are displayed by the following proposition.

Proposition 4.1.1 (Conjugate Analysis). *Let the prior information over μ, η be modelled with a Normal Inverted Gamma Model ($[M, H] \sim \mathcal{NIG}(\alpha, \beta, m, \Lambda)$) and let the likelihood follow a normal*

¹Note that some authors use $\frac{1}{\beta}$ as the second parameter for the inverse gamma distribution [4].

distribution ($D \sim \mathcal{N}(\mu, \sqrt{\eta})$). Then the posterior distribution is again Normal Inverted Gamma distributed, and moreover, the parameters of the posterior distribution are explicitly given as $([M, H]|[D] \sim \mathcal{NIG}(\alpha^*, \beta^*, m^*, \Lambda^*))$. These parameter can be expressed in terms of the parameters of the prior information as follows:

$$\Lambda^* = \frac{\Lambda}{1 + \Lambda}, \quad m^* = \frac{m + \delta\Lambda}{1 + \Lambda}, \quad \alpha^* = \alpha + \frac{1}{2}, \quad \beta^* = \frac{2\beta(1 + \Lambda)}{\beta(m - \delta)^2 + 2(1 + \Lambda)}. \quad (4.1.4)$$

Proof. ² Bayes theorem states the following property for the posterior density

$$f_{M,H|D}(\mu, \eta; \delta) = C_1 L(\delta; \mu, \eta) f_{M,H}(\mu, \eta), \quad (4.1.5)$$

where C_1 is a normalizing constant such that the posterior distribution is proper, and L is the likelihood of the data. Substituting the distribution functions in (4.1.5) results in

$$f_{M,H|D}(\mu, \eta; \delta) = C_1 \frac{1}{\sqrt{2\pi\eta}} e^{-\frac{1}{2\eta}(\delta - \mu)^2} \frac{\eta^{-\alpha - \frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1}{2\eta\Lambda}(\mu - m)^2 - \frac{1}{\eta\beta}]}. \quad (4.1.6)$$

This equation can be further reduced as

$$\begin{aligned} f_{M,H|D}(\mu, \eta; \delta) &= C_1 \frac{\eta^{-\alpha - 2} \beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1}{2\eta\Lambda}(\Lambda(\delta - \mu)^2 + (\mu - m)^2) - \frac{1}{\eta\beta}]} \\ &= C_1 \frac{\eta^{-\alpha - 2} \beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1}{2\eta\Lambda}(\mu^2[1 + \Lambda] - 2\mu[\delta\Lambda + m] + m^2 + \Lambda\delta^2) - \frac{1}{\eta\beta}]} \\ &= C_1 \frac{\eta^{-\alpha - 2} \beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1 + \Lambda}{2\eta\Lambda}(\mu - \frac{\delta\Lambda + m}{1 + \Lambda})^2 + \frac{(1 + \Lambda)(\delta\Lambda + m)^2}{2\eta\Lambda(1 + \Lambda)^2} - \frac{1}{2\eta\Lambda}(m^2 + \Lambda\delta^2) - \frac{1}{\eta\beta}]}. \end{aligned} \quad (4.1.7)$$

Re-parametrization with $m^* = \frac{\delta\Lambda + m}{1 + \Lambda}$ and $\Lambda^* = \frac{\Lambda}{1 + \Lambda}$ results in

$$f_{M,H|D}(\mu, \eta; \delta) = C_1 \frac{\eta^{-\alpha - 2} \beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1}{2\eta\Lambda^*}(\mu - m^*)^2 + \frac{m^{*2}}{2\eta\Lambda^*} - \frac{m^2 + \Lambda\delta^2}{2\eta\Lambda} - \frac{1}{\eta\beta}]}. \quad (4.1.8)$$

By defining $\frac{1}{\beta^*} = \frac{1}{\beta} + \frac{(m - \delta)^2}{2(1 + \Lambda)}$ and substituting this in (4.1.8) leads to

$$f_{M,H|D}(\mu, \eta; \delta) = C_1 \frac{\eta^{-\alpha^* - 2} \beta^{-\alpha^*}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{[-\frac{1}{2\eta\Lambda^*}(\mu - m^*)^2 - \frac{1}{\eta\beta^*}]}. \quad (4.1.9)$$

Also define $\alpha^* = \alpha + \frac{1}{2}$. Recall that this is a density function defining the last constant C_1 .

Concluding, the following posterior density function is found

$$f_{M,H|D}(\mu, \eta; \delta) = \frac{\eta^{-\alpha^* - \frac{3}{2}} \beta^{-\alpha^*}}{\sqrt{2\pi}\Gamma(\alpha^*)\Lambda^{*\frac{1}{2}}} e^{-\frac{1}{2} \frac{(\mu - m^*)^2}{\eta\Lambda^*} - \frac{1}{\eta\beta^*}}, \quad (4.1.10)$$

which is clearly that of $\mathcal{NIG}(\alpha^*, \beta^*, m^*, \Lambda^*)$. Notice that the dependence of the posterior on the data is fully expressed through the parameters (m^*, β^*) of the posterior distribution. \square

²For another version of the proof see [4].

The previous stated proposition showed and proved the conjugate properties when using the \mathcal{NIG} -model. Note that this was only for the update step (inversion) and not (yet) for the predictive (forward) step which is discussed in Section 5.1.

4.2 Direct Numerical Integration Approach

A different approach that can be taken to perform the updating procedure (step three in the Bayesian method) is *direct numerical integration*. The name refers to the procedure used to tackle the integration derived when performing the update. This can involve calculating the normalizing constant (see Equation (1.2.4)) or performing the integration needed in applying LTP. The most commonly used integration method is direct numerical integration. This means that the integration is performed by calculating the upper (Riemann) sum of the integral. Let $x_i, i = 0 \dots n$ form a partition of the region $[a, b]$. Then the following approximation holds for n sufficiently large (or $\Delta x_i = x_i - x_{i-1}$ sufficient small)

$$\int_a^b f(x)dx = \sum_{i=1}^n f(x_i)\Delta x_i. \quad (4.2.1)$$

An example of this approach can be found in the Appendix C as `numbTimbN.m`. Direct numerical integration is a powerful tool as it is very robust to the models used in modelling the data and prior information, e.g., no major restrictions on the family of the distributions. This as the distribution functions are only used as functional relations. The only problem with this method is that it uses a lot of computation as a discretisation needs to be defined over all variables which are used in the computation. This makes this approach hard to perform when many variables are needed to perform the Bayesian method, which will be displayed in Section 5.2.

4.3 BBN Approach

The main theory underlying this approach is that of (discrete) (Bayesian) belief nets(BBNs)³. In this subsection they are discussed in the framework of a software package that can handle BBNs, called HUGIN. Using this package is not necessary for this approach, for more information on *HUGIN* see their internet site at <http://www.hugin.com/>. The belief net consists of two parts, namely a graphical and quantitative part. First the graphical part is discussed. This is a graphical representation of the underlying relations. The underlying principles of constructing a belief net will be shortly discussed here, for a full discussion of the topic see [12]. In the graphical representation of the belief net, the random variables are displayed with nodes. The relation between nodes is modelled by a directed arc. No cycles are allowed in the graphical model⁴. The following figure shows the graphical part of the belief net for the simplified timber case.

Notice that the R is influenced by MU. Therefore the arc is directed from MU towards R. Similarly the arc is drawn from MU to D(the destructive measurement). Note that this also shows that D and R are conditionally independent given MU. Now that the relations between nodes are modelled the second part of the belief net referred to as the quantitative modelling part needs to be defined. This part consist of conditional probability tables for all the nodes conditional on their parent nodes. The

³In this thesis only the use of discrete BBNs is discussed.

⁴Using causal relations between the variables to define the links between the nodes, will diminish the chance of creating cycles.

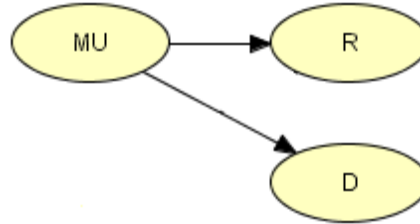


Figure 4.1: General belief net representation for the timber case

parent nodes are defined as the nodes connected via incoming arcs. For example MU is called the parent node of R⁵, this as there exists an incoming arc at R from MU. These conditional probability tables (CPTs) are (in case of a discrete BBN) discrete version of the conditional distribution function conditional on the parent nodes. For a prime node⁶ this CPT shows the prior distribution over this parameter.

In HUGIN all these probabilities in the CPT need to by default be entered by hand. With a large number of nodes or large discretisation this can become a cumbersome job. Therefore HUGIN also lets you define it with a functional relation(model) which can only depend on parent nodes and constants. There are some restrictions to the use of these models. HUGIN only supports a limited number of distributions and only certain operators are allowed⁷. Luckily HUGIN can handle the normal distribution thus the CPTs needed when using the \mathcal{N} -model can be generated with the help of these models.

Note that the graphical HUGIN model is almost the same graphical representation given in Figure 3.4, the main difference is that in HUGIN variables are only shown in the graphical model and the models are used to define the qualitative part of the network.

After the model is defined it can be used to compute the posterior (step three in the Bayesian method) and make inferences (Step four). Note that defining the entire model is in fact implementing all the ingredients needed for Bayes rule, i.e. the first two steps of the Bayesian method. HUGIN can now compile the model, which means performing the update. After compilation of the BBN the evidence (measurements) can be entered by selecting a state of a node. HUGIN will propagate this evidence through the network and all distributions are updated. This is essentially applying Bayes Rule. In Figure 4.2 the compiled network for the simplified timber case is displayed. On the left in this figure the prior distribution are shown. On the right the evidence is entered by selecting a state, indicated by the red highlighting. When comparing the two distributions for MU (prior and posterior) in Figure 4.2 it is clearly seen that the distribution of MU really shifts when entering the evidence in the network.

Notice that HUGIN can only give marginal posteriors. As can also be seen from this small example, HUGIN is an easy to use tool when performing Bayesian updates. The graphical model clearly displays the relations between the variables, and the update can be performed for different data reasonably fast. There are some small disadvantages when using HUGIN and (discrete) BBN in general. The main problem using a discrete BBN is defining the needed discretisation. This can sometimes be a difficult task. Also the results from HUGIN are not very easily post-processed.

⁵R is called the child node of MU.

⁶A prime node is a node with no parents.

⁷The HUGIN Manual [11] constitutes a complete list of all the distributions and operators available.

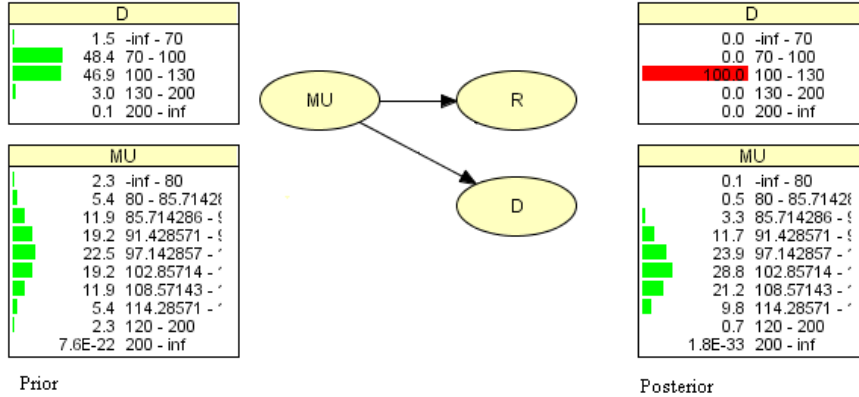


Figure 4.2: Prior and posterior distribution in HUGIN

A lot of problems encountered using HUGIN can be overcome when using the HUGIN API. For instance it allows control over HUGIN via MATLAB. This makes it possible to implement functions to define the discretisations or probabilities and it also makes post-processing of the results easier. In Appendix B a report can be found that explains how to control HUGIN via this API. This report contains some examples in using the API and explains the syntax needed to use the API in MATLAB.

4.4 Bayes Linear Approach

Bayes linear is another method that was proposed to tackle updating. This approach was investigated as it posed to be very fast in terms of computation-time. The drawback was that only the updated first two moments would be computed and not the entire distribution. Under the assumption that the posterior distribution would still follow a normal distribution this is no drawback at all. The methodology of Bayes linear has been developed by Goldstein see [7], [15], and [16].

When using Bayes linear the steps of the Bayesian method (Definition (3.1.1)) are as follows. In step one the prior information needs to be defined by stating the first two moments of the prior distribution, $\mathbb{E}(\Theta)$, $\text{Var}(\Theta)$. In step two the likelihood needs to be defined, this is done by stating the first two moments of the data(X) and the covariance of the data and the quantity inference needs to be made upon, $\mathbb{E}(X)$, $\text{Var}(X)$, $\text{Cov}(X, \Theta)$. The updating⁸ defined as step three is given by the following two equations. They give the adjusted first two moments by Bayes linear:

$$\mathbb{E}_X(\Theta) = \mathbb{E}(\Theta) + \text{Cov}(\Theta, X) (\text{Var}(X))^{-1} [X - \mathbb{E}(X)] \quad (4.4.1)$$

$$\text{Var}_X(\Theta) = \text{Var}(\Theta) - \text{Cov}(\Theta, X) (\text{Var}(X))^{-1} \text{Cov}(X, \Theta). \quad (4.4.2)$$

The first equation can be derived by minimizing the following expression (quadratic loss)

$$\mathbb{E}(\Theta - \mathbb{E}_X(\Theta))^2. \quad (4.4.3)$$

⁸The term adjusted is used in Bayes linear and represents the same as updated.

The second can be found by setting the quadratic loss, Equation (4.4.3) with the adjusted expectation, Equation (4.4.1) substituted into it to the adjusted variance. Now step four can be performed, making inference by comparing the adjusted quantities with the prior first two moments. This will be demonstrated by applying it to the simplified timber case.

In the simplified timber case the strength of the timber is modelled as being normally distributed with the following parameters, unknown M and fixed σ , thus

$$R|(M, \sigma) \sim \mathcal{N}(M, \sigma). \quad (4.4.4)$$

The prior distribution over M is assumed to be normal with parameters μ_M, σ_M . This means in the Bayes linear approach that the prior is given as

$$\mathbb{E}(M) = \mu_M, \text{Var}(M) = \sigma_M^2. \quad (4.4.5)$$

The second step defining the likelihood is first performed for a single observation. For a single destructive observation (D) the likelihood in terms of Bayes linear is stated by $\mathbb{E}(D)$, $\text{Var}(D)$, $\text{Cov}(M, D)$. These properties can be calculated by using the definition which implies evaluating an integral (at most two-dimensional). This is shown for $\text{Cov}(M, D)$. By definition the covariance can be calculated with the following formula

$$\text{Cov}(M, D) = \mathbb{E}(MD) - \mathbb{E}(M)\mathbb{E}(D). \quad (4.4.6)$$

The first term on the rhs. of Equation (4.4.6) is calculated as follows.

$$\begin{aligned} \mathbb{E}(MD) &= \iint \mu \delta f_{M,D}(\mu, \delta) d(\delta, \mu) \\ &= \iint \mu \delta f_{D|M}(\delta; \mu) f_M(\mu) d(\delta, \mu) \\ &= \int \mu f_M(\mu) \int \delta f_{D|M}(\delta; \mu) d\delta d\mu \\ &= \int \mu f_M(\mu) \mu d\mu \\ &= \sigma_M^2 + \mu_M^2 \end{aligned} \quad (4.4.7)$$

From these calculations it follows that

$$\text{Cov}(M, D) = \sigma_M^2, \mathbb{E}(D) = \mu_M, \text{Var}(D) = \sigma^2 + \sigma_M^2. \quad (4.4.8)$$

Substituting Equation (4.4.8) into the general expressions for the adjusted quantities, Equations (4.4.1), (4.4.2) concludes step three of the Bayesian method, which results in

$$\mathbb{E}_D(M) = \mu_M + \sigma_M^2 (\sigma^2 + \sigma_M^2)^{-1} [\delta - \mu_M] \quad (4.4.9)$$

$$= \frac{\sigma_M^2 \delta + \sigma^2 \mu_M}{\sigma_M^2 + \sigma^2}, \quad (4.4.10)$$

$$\text{Var}_D(M) = \sigma_M^2 - \sigma_M^2 (\sigma^2 + \sigma_M^2)^{-1} \sigma_M^2 \quad (4.4.11)$$

$$= \frac{\sigma^2 \sigma_M^2}{\sigma_M^2 + \sigma^2}. \quad (4.4.12)$$

Remark 4.4.1. The adjusted first two moments given by Bayes linear are exactly the same in the simplified timber using destructive data as when applying the \mathcal{N} -model and the analytical or direct numerical integration approach. See for instance Equation (3.2.5).

Inference can be made on the variable $M(\mu)$ by comparing the prior and adjusted first two moments. Also a predictive distribution can be calculated if a posterior normal distribution shape is assumed for the parameter $M(\mu)$. This is done via applying LTP which can be done using an analytical solution or direct numerical integration.

If more destructive observations are used in the Bayes linear approach the quantities in Equations (4.4.1), (4.4.2) need to be interpreted as vectors and matrices. This implies that in step two of the Bayesian method, which is define the likelihood the Bayes linear equivalent requires to also calculate $\text{Cov}(D_i, D_j)$, for $i \neq j$, which is the covariance between different destructive observations. As the reader can easily verify, the adjusted first two moments using n destructive observations are equal to

$$\mathbb{E}_{\underline{D}}(M) = \mu_M + \sigma_M^2 \mathbf{1}_{1,n} [\mathbf{Diag}_n(\sigma^2) + \sigma_M^2 \mathbf{1}_{n,n}]^{-1} [\underline{d} - \mu_M \mathbf{1}_{n,1}], \quad (4.4.13)$$

$$\text{Var}_{\underline{D}}(M) = \sigma_M^2 - \sigma_M^2 \mathbf{1}_{1,n} [\mathbf{Diag}_n(\sigma^2) + \sigma_M^2 \mathbf{1}_{n,n}]^{-1} \sigma_M^2 \mathbf{1}_{n,1}. \quad (4.4.14)$$

Where $\mathbf{1}_{n,m}$, $\mathbf{Diag}_n(k)$ denotes respectively a matrix of size (n, m) with all ones as entries and the diagonal matrix of size (n, n) with a vector of k 's on the main diagonal.

If non-destructive observations, $\underline{\Upsilon}(v)$ are used the likelihood changes. In case of using an additive error model to model the measurement error the changes are relatively easy to calculate. This is due to the linearity of the covariance, variance and expectation. The likelihood step in Bayes linear then becomes

$$\text{Cov}(M, \Upsilon) = \text{Cov}(M, R + \varepsilon) = \text{Cov}(M, R) + \text{Cov}(M, \varepsilon) = \text{Cov}(M, R) = \sigma_M^2, \quad (4.4.15)$$

$$\text{Var}(\Upsilon) = \text{Var}(R + \varepsilon) = \text{Var}(R) + \text{Var}(\varepsilon) = \sigma^2 + \sigma_M^2 + \sigma_\varepsilon^2. \quad (4.4.16)$$

The adjusted first two moments when using p non-destructive observations with an additive error-model equal:

$$\mathbb{E}_{\underline{\Upsilon}}(M) = \mu_M + \sigma_M^2 \mathbf{1}_{1,p} [\mathbf{Diag}_p(\sigma^2 + \sigma_\varepsilon^2) + \sigma_M^2 \mathbf{1}_{p,p}]^{-1} [\underline{v} - \mu_M \mathbf{1}_{p,1}], \quad (4.4.17)$$

$$\text{Var}_{\underline{\Upsilon}}(M) = \sigma_M^2 - \sigma_M^2 \mathbf{1}_{1,p} [\mathbf{Diag}_p(\sigma^2 + \sigma_\varepsilon^2) + \sigma_M^2 \mathbf{1}_{p,p}]^{-1} \sigma_M^2 \mathbf{1}_{p,1}. \quad (4.4.18)$$

Using the multiplicative model to model the measurement error is also possible when using the Bayes linear approach. However when doing so the calculations become a bit more comprehensive, that is why it is not displayed here.

The use of both destructive and non-destructive data have been discussed, and the Bayes linear approach applied to the simplified timber case can be defined. First define the following matrices.

$$\text{Data} = \begin{bmatrix} \underline{D} & \underline{\Upsilon} \end{bmatrix}^T \quad (4.4.19)$$

$$\text{Cov}(M, \text{Data}) = \begin{bmatrix} \text{Cov}(M, \underline{D}) & \text{Cov}(M, \underline{\Upsilon}) \end{bmatrix} \quad (4.4.20)$$

$$\text{Var}(\text{Data}) = \begin{bmatrix} \text{Var}(\underline{D}) & \text{Cov}(\underline{\Upsilon}, \underline{D}) \\ \text{Cov}(\underline{D}, \underline{\Upsilon}) & \text{Var}(\underline{\Upsilon}) \end{bmatrix} \quad (4.4.21)$$

The adjusted first two moments are given by substituting the above matrices in Equations (4.4.1), (4.4.2), this has been implemented in MATLAB (see Appendix C). In Figure 4.3 we see a plot of the prior and the posterior produced by these m-files. In order to compare the calculated posterior of Bayes linear to the posterior calculated by direct numerical integration (with the \mathcal{N} -model) have both have been plotted.

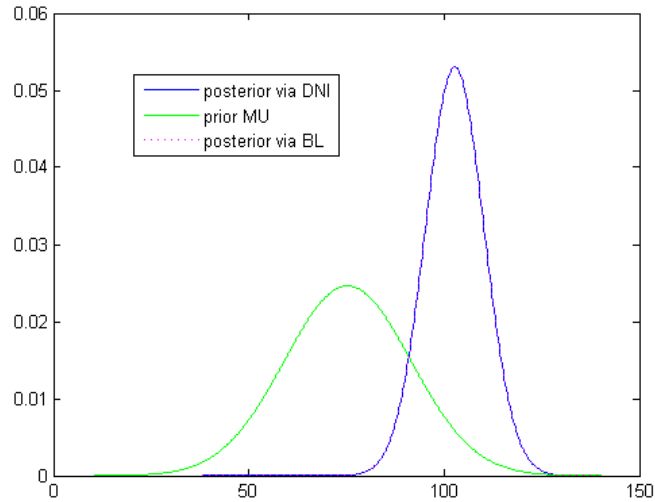


Figure 4.3: A plot of the posterior distribution of M via Bayes linear and direct numerical integration in the simplified timber case

As can be seen in Figure 4.3 both the approaches lead to the same posterior distribution of $M(\mu)$, which was also shown by the previous calculations. The computation time when using Bayes linear approach is smaller than when using direct numerical integration approach. In situation investigated in this thesis research on average a difference of 25 percent were observed.

This concludes the discussion of Bayes linear applied to the simplified timber case. In Appendix A.2, Bayes linear is applied to the general timber case and compared to the other approaches. From which it is concluded that the second moment is not correct in general when using Bayes linear.

Chapter 5

Application of the Different Approaches to the Timber Case

This chapter describes some advanced applications of the Bayesian method. These applications were all encountered whilst working on the timber case. If no data-set is mentioned the data-set ‘Camaru’ is used. This data-set contains five destructive and five non-destructive test results on the timber called camaru. The following tables display the used data-set and the parameters for the prior information, respectively.

Table 5.1: Test results for Camaru timber ($[N/mm^2]$)

Destructive Tests	Non-Destructive Tests
121.3144687	105.0329832
112.5568927	102.8546259
111.8634926	103.9495366
98.78462057	94.07234876
158.2366957	104.8491008

Table 5.2: Parameters prior distributions

	Mean	Std. Dev.
M	75.29	23.32
Σ	16.2	4.36
ϵ	1	0.22

5.1 Application of the Analytical Approach

The \mathcal{NIG} -model has been derived in Section 4.1. In this section it will be applied to the timber case. The \mathcal{NIG} -model is used to model the priors on $M(\mu)$ and $H(\eta)$. Notice that this last statement is equivalent to setting a prior on σ , because $\sigma^2 = \eta$, which was done in the previous chapters.

This section shows the advantages of using a conjugate analysis. The down side of using a conjugate model is the restriction that is put on the distributions that can be used. These restrictions do not always make the model applicable in certain cases as validation with the real process it tries to model fails. For instance when using the normal model. The normal model is used very often as it is well known and thus often a natural choice. A drawback of using the normal model is that it is a symmetric distribution and the process it tries to model is not. The following example also shows problems encountered during the research of this thesis. When using the normal model to model the distribution of a standard deviation (as a random variable itself) there was density mass placed at negative values. By definition the standard deviation cannot be negative but applying the normal model with a small mean and large variance as a prior for the standard deviation does implicate this. Which makes applying the \mathcal{N} -model in this case not correct.

The remainder of this subsection describes the calculations needed when using the \mathcal{NIG} -model to perform a conjugate analysis in the timber case.

One destructive observation

In this setting only the outcome of one destructive test will be used to update the parameters of the distribution of R . Thus the parameters for the predictive distribution imposed by the use of the \mathcal{NIG} -model depend only on one data measurement and the prior information. The result of applying the \mathcal{NIG} -model is stated in the following proposition.

Proposition 5.1.1. *Let $R \sim \mathcal{N}(\mu, \sqrt{\eta})$, and let the prior distributions for $M, H(\mu, \eta)$ be defined as $[M, H] \sim \mathcal{NIG}(\alpha, \beta, m, \Lambda)$, and $D \sim \mathcal{N}(\mu, \sqrt{\eta})$. Then the predictive distribution for R is (generalized) student's t -distributed with density function*

$$f_{R|D}(r; \delta^*) = \frac{\beta^{*-\alpha^*} \Gamma(\alpha^* + \frac{1}{2})}{\sqrt{2\pi(1 + \Lambda^*)} \Gamma(\alpha^*)} (-C_2)^{-\alpha^* - \frac{1}{2}}, \quad (5.1.1)$$

with

$$C_2 = -\frac{1}{\beta^*} - \frac{(r - m^*)^2}{2(1 + \Lambda^*)}, \quad (5.1.2)$$

and

$$\Lambda^* = \frac{\Lambda}{1 + \Lambda}, \quad m^* = \frac{m + \delta\Lambda}{1 + \Lambda}, \quad \alpha^* = \alpha + \frac{1}{2}, \quad \beta^* = \frac{2\beta(1 + \Lambda)}{\beta(m - \delta)^2 + 2(1 + \Lambda)}. \quad (5.1.3)$$

Proof. By Proposition 4.1.1 the posterior distribution of M, Σ is again \mathcal{NIG} -distributed with parameters $\alpha^*, \beta^*, m^*, \Lambda^*$. Applying LTP results in

$$\begin{aligned}
f_{R|D}(r; \delta) &= \iint \frac{1}{\sqrt{2\pi\eta}} e^{[-\frac{1}{2\eta}(r-\mu)^2]} \frac{\eta^{-\alpha^* - \frac{3}{2}} \beta^{*- \alpha^*}}{\sqrt{2\pi\Gamma(\alpha^*)\Lambda^{*\frac{1}{2}}}} e^{\left[-\frac{1}{2} \frac{(\mu-m^*)^2}{\eta\Lambda^*} - \frac{1}{\eta\beta^*}\right]} d(\mu, \eta) \\
&= \iint \frac{\eta^{-\alpha^* - 2} \beta^{*- \alpha^*}}{2\pi\Gamma(\alpha^*)\Lambda^{*\frac{1}{2}}} e^{-\left[\frac{1}{\eta\beta^*}\right]} e^{-\frac{1}{2\eta\Lambda^*} [\Lambda^*(r-\mu)^2 + (\mu-m^*)^2]} d(\mu, \eta) \\
&= \int \frac{\eta^{-\alpha^* - 2} \beta^{*- \alpha^*}}{2\pi\Gamma(\alpha^*)\Lambda^{*\frac{1}{2}}} e^{-\left[\frac{1}{\eta\beta^*}\right]} e^{-\frac{(r-m^*)^2}{2\eta(\Lambda^*+1)}} d\eta \sqrt{\frac{2\pi\eta\Lambda^*}{1+\Lambda^*}} \\
&= \frac{\beta^{*- \alpha^*}}{\sqrt{2\pi(1+\Lambda^*)\Gamma(\alpha^*)}} \int \eta^{-\alpha^* - \frac{3}{2}} e^{-\frac{C_2}{\eta}} d\eta,
\end{aligned}$$

where

$$C_2 = -\frac{1}{\beta^*} - \frac{(r-m^*)^2}{2(1+\Lambda^*)}. \quad (5.1.4)$$

When changing the integration variables from η to $\xi = -\frac{C_2}{\eta}$ transforms the integral to a gamma function,

$$\begin{aligned}
f_{R|D}(r; \delta) &= \frac{\beta^{*- \alpha^*}}{\sqrt{2\pi(1+\Lambda^*)\Gamma(\alpha^*)}} \int \left(-\frac{C_2}{\xi}\right)^{-\alpha^* - \frac{3}{2}} e^{-\xi} \left(-\frac{C_2}{\xi^2}\right) d\xi \\
&= \frac{\beta^{*- \alpha^*} (-C_2)^{-\alpha^* - \frac{1}{2}}}{\sqrt{2\pi(1+\Lambda^*)\Gamma(\alpha^*)}} \int \xi^{\alpha^* - \frac{1}{2}} e^{-\xi} d\xi \\
&= \frac{\beta^{*- \alpha^*} (-C_2)^{-\alpha^* - \frac{1}{2}}}{\sqrt{2\pi(1+\Lambda^*)\Gamma(\alpha^*)}} \Gamma\left(\alpha^* + \frac{1}{2}\right). \quad (5.1.5)
\end{aligned}$$

The predictive distribution of R has the following density function

$$f_{R|D}(r; \delta) = \frac{\beta^{*- \alpha^*} \Gamma\left(\alpha^* + \frac{1}{2}\right)}{\sqrt{2\pi(1+\Lambda^*)\Gamma(\alpha^*)}} (-C_2)^{-\alpha^* - \frac{1}{2}}. \quad (5.1.6)$$

Noting that this is recognized as a (generalized) student's t-distribution, concludes the proof. Also note that the dependence of the data, $D(\delta)$ is completely fixed in the parameters of the posterior distribution of M, Σ , i.e. , in β^*, m^* . \square

Multiple destructive observations

This subsection will tackle the timber case when more than one destructive test measurement is used in the updating procedure. This case can be dealt with briefly, because it leads to the same procedure as the one-dimensional case presented in the previous subsection. Due to the conditional independence given (M, H) of the destructive measurements in the timber case, the likelihood is a

product of the normal density functions which implies that the likelihood itself is again normally distributed. Thus Proposition 5.1.1 can be used. Notice that only the value of the parameters in the distribution are different, they are dependent on more measurements.

Proposition 5.1.2. *The predictive distribution for R determined with n destructive measurements using \mathcal{NIG} -model is (generalized) student's t -distributed where the density function is given by Equation (5.1.1) and (5.1.2) with parameters:*

$$\Lambda^* = \frac{\Lambda}{1+n\Lambda}, m^* = \frac{m + \Lambda \sum \delta_i}{1+n\Lambda}, \alpha^* = \alpha + \frac{n}{2}, \quad (5.1.7)$$

$$\beta^* = \frac{2\beta(1+n\Lambda)}{-\beta(\Lambda(\sum \delta_i)^2 + 2\sum \delta_i m - nm^2 - (1+\Lambda n)\sum \delta_i^2) + 2(1+n\Lambda)}.$$

Proof. The proof of this proposition is added in the Appendix A.1. It follows the lines of the proofs of propositions 5.1.1 and 4.1.1. □

Non-destructive observations

The \mathcal{NIG} -model will be applied to the non-destructive data to derive the predictive distribution. Again first one measurement from a non-destructive test is used. The error is assumed to be additive, see Equation (2.0.7). The main difference in using non-destructive data instead of destructive data is with the likelihood, $L(v; \mu, \eta)$. In order to determine the likelihood function the LTP is used in the following way.

$$\begin{aligned} L(v; \mu, \eta) &= \int f_{Y|M,H,\varepsilon}(v; \mu, \eta, a) f_\varepsilon(a) da \\ &= \int f_{R|M,H}(v-a; \mu, \eta) f_\varepsilon(a) da \\ &= \int \frac{1}{\sqrt{2\pi\eta}} e^{-\frac{1}{2\eta}(v-a-\mu)^2} \frac{1}{\sqrt{2\pi\sigma_\varepsilon}} e^{-\frac{1}{2\sigma_\varepsilon^2}(a-\mu_a)^2} da \end{aligned}$$

This integral can be computed via ‘completing the square’. This results in the following likelihood function

$$L(v; \mu, \eta) = \frac{1}{\sqrt{2\pi(\sigma_\varepsilon^2 + \eta)}} e^{-\frac{(v-\mu-\mu_a)^2}{\sigma_\varepsilon^2 + \eta}}. \quad (5.1.8)$$

Notice that this equation represents a normal distribution, $\mathcal{N}(\mu + \mu_a, \sqrt{\sigma_\varepsilon^2 + \eta})$. This makes the \mathcal{NIG} -model not directly applicable as now the conjugate properties are not present. Therefore the following transformation is used

$$\tilde{\eta} = \eta + \sigma_\varepsilon^2. \quad (5.1.9)$$

Notice that the likelihood of the data then becomes a normal distribution with variance \tilde{H} . The prior for M, \tilde{H} equals

$$\begin{aligned} f_{M, \tilde{H}}(\mu, \tilde{\eta}) &= f_{M, H}(\mu, \eta + \sigma_\varepsilon^2) \\ &= \frac{(\eta + \sigma_\varepsilon^2)^{-\alpha - \frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi} \Gamma(\alpha) \Lambda^{\frac{1}{2}}} e^{-\frac{1}{2} \frac{(\mu - m)^2}{(\eta + \sigma_\varepsilon^2) \Lambda} - \frac{1}{(\eta + \sigma_\varepsilon^2) \beta}}, \end{aligned} \quad (5.1.10)$$

which is recognized as a $\mathcal{NIG}(\alpha, \beta, m, \Lambda)$ -distribution in terms of (M, \tilde{H}) . This means that a conjugate analysis is possible on the random variables (M, \tilde{H}) . This implies that the posterior distribution of (M, \tilde{H}) is again \mathcal{NIG} -distributed with the following parameters $(\alpha^*, \beta^*, m^*, \Lambda^*)$

$$\alpha^* = \alpha + \frac{1}{2}, \quad (5.1.11)$$

$$\beta^* = \frac{2\beta(1 + \Lambda)}{\beta(v - \mu_\varepsilon - m)^2 + 2(1 + \Lambda)}, \quad (5.1.12)$$

$$m^* = \frac{\Lambda(v - \mu_\varepsilon) + m}{1 + \Lambda}, \quad (5.1.13)$$

$$\Lambda^* = \frac{\Lambda}{1 + \Lambda}. \quad (5.1.14)$$

This implies that the posterior distribution for (M, H) is given by this \mathcal{NIG} -distribution transformed by the transformation given as Equation (5.1.9). In order to calculate the predictive distribution of R also the posterior distribution of (M, \tilde{H}) is used, this cannot be done analytically.

$$f_{R|\Upsilon}(r; v) = \iiint f_{R|M, \tilde{H}}(r, \mu, \tilde{\eta}) f_{M, \tilde{H}|\Upsilon}(\mu, \tilde{\eta}; v) d(\mu, \tilde{\eta}) \quad (5.1.15)$$

The predictive distribution of R conditional on (M, H) is normal with parameters (M, H) . Therefore by relation (5.1.9), the distribution of R conditional on (M, \tilde{H}) is also normal with parameters $(M, \tilde{H} - \sigma_\varepsilon^2)$.

Examples \mathcal{NIG} -model

In the previous subsection the application of the \mathcal{NIG} -model to the timber case was discussed in a general setting with explicit prior information. This subsection will discuss the use of vague prior information and will give an example using an actual data-set. In order to use the \mathcal{NIG} -model to model the two-dimensional prior over μ and $\eta = \sigma^2$ four parameters need to be defined, α, β, m and Λ respectively the scale and shape parameters of the inverse gamma distribution over η , and parameters of the normal distribution over μ .

As mentioned earlier in this thesis in many cases prior information is not known. In such cases a vague/non informative prior is often used. In the normal case it can be shown that an appropriate choice is to use σ^{-1} for Σ . This non informative prior can also be used in the \mathcal{NIG} -model. It can be derived as a limiting case of the \mathcal{NIG} -model, set the parameter $\Lambda = \infty$. Then

$$f_{M, H}(\mu, \eta) \propto \eta^{-\alpha - \frac{3}{2}} e^{\frac{1}{\eta\beta}}. \quad (5.1.16)$$

By setting the values $\alpha = -1$ and $\beta = \infty$ the following prior is found

$$f_{M, H}(\mu, \eta) \propto \eta^{-\frac{1}{2}}. \quad (5.1.17)$$

Notice that this is an improper prior, therefore the α -sign was used. This concludes the discussion on the case of vague prior information in the \mathcal{NIG} -model.

If prior information is available it can be included in the \mathcal{NIG} -model. This needs to be done by fitting the four parameters of the model. This is usually done by fitting the marginal distributions of the model. The parameters (α, β) should be chosen such that the prior marginal distribution of η reflects the prior information in η . The parameters m, Λ have to be selected m, Λ such that the marginal distribution of μ reflects the prior information on μ . These marginal distributions can be found by integrating out one variable in the joint prior density function. Note that by using the \mathcal{NIG} -model a fixed dependence is assumed on M, H .

When the prior information is only defined by its first two moments (Bayes linear) the following set of equations can be derived. They define the parameters in terms of the first two moments.

$$\mathbb{E}(\mu) = m, \quad (5.1.18)$$

$$\mathbb{E}(\eta) = \frac{1}{(\alpha - 1)\beta}, \quad (5.1.19)$$

$$\text{Var}(\mu) = \frac{1}{(\alpha - 1)\beta} \Lambda, \quad (5.1.20)$$

$$\text{Var}(\eta) = \frac{1}{(\alpha - 1)^2(\alpha - 2)\beta^2}. \quad (5.1.21)$$

Example 5.1.1. *The following special setting in the timber case is examined. Let the prior information over the strength of this timber be modelled with the following parameters in the \mathcal{NIG} -model*

$$\alpha = 9, \beta = \frac{1}{100}, m = 100, \Lambda = 8. \quad (5.1.22)$$

Which means by Equations (5.1.19), (5.1.20), (5.1.21), and (5.1.21) that the prior information can be summarized as

$$\mathbb{E}(\mu) = 100, \text{Var}(\mu) = 100, \mathbb{E}(\eta) = \frac{100}{8}, \text{and } \text{Var}(\eta) = 22.3. \quad (5.1.23)$$

Thus the strength of a beam is a-priori expected to be a hundred [N/mm^2]. Figure 5.1 displays this prior information.

Now two random selected beams from this population of timber are each destructively tested. These tests provide in total two individual measurements of the strength of those two specific beams.

Table 5.3: Observed destructive strengths

Test Nr.	Strength [N/mm^2]
1	120
2	110

This data and the prior information are used in Equation (5.1.1) to perform the updating. The calculations needed to perform the conjugate analysis were implemented in MATLAB¹ and the results are plotted in Figure 5.2. Notice that a student's t -type posterior is displayed in this figure. Which was proven and defined by Theorem 5.1.1. From this figure it can also be concluded that the used observations yield a higher value for the mean strength of the timber. Another result of using the observations is that the variance of the strength of the timber has been reduced. This can be interpreted as diminishing the uncertainty.

¹The m -files that were used can be found in Appendix C. Also some additional files for handling the \mathcal{NIG} -model in MATLAB are listed.

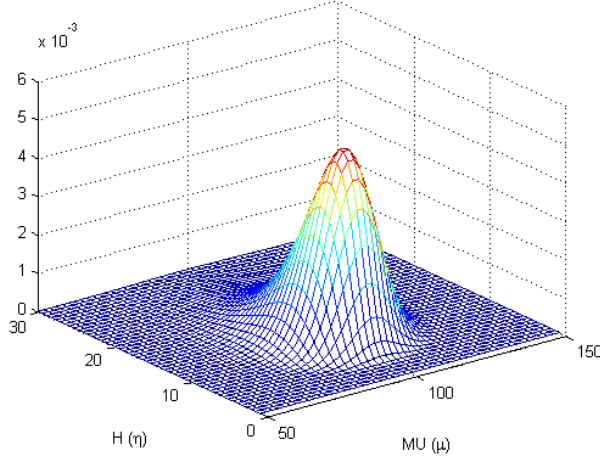


Figure 5.1: A plot of the prior information modelled by a normal inverted gamma density

5.2 Application of the Direct Numerical Integration Approach

Direct numerical integration is going to be used with the \mathcal{N} -model and an improper prior. This means that first a normal distribution is set as a prior distribution on $M(\mu)$ and $\Sigma(\sigma)^2$.

$$\begin{aligned} f_{M,\Sigma}(\mu, \sigma) &= f_M(\mu)f_\Sigma(\sigma) \\ &= \frac{1}{\sigma_\mu} \varphi\left(\frac{\mu - \mu_\mu}{\sigma_\mu}\right) \frac{1}{\sigma_\sigma} \varphi\left(\frac{\sigma - \mu_\sigma}{\sigma_\sigma}\right). \end{aligned} \quad (5.2.1)$$

Note that when using this prior, a-priori M and Σ are modelled as independent, and that a conjugate analysis is no longer possible³.

Using this approach in the timber case three kinds⁴ of data from the timber case can be handled, as mentioned in Chapter 2 only two will be dealt with, namely destructive (D) and non-destructive (Υ) data. In case of destructive data the likelihood (which is determines the data) is given as a normal distribution with parameters (the unknown) M, Σ . In case of multiple destructive measurements the likelihood is the product of the individual likelihoods, as all measurements are conditional independent given the parameters of the distribution. The likelihood (for n destructive measurements) equals

$$L_{\underline{D}}(\underline{d}) = f_{\underline{D}|M,\Sigma}(\underline{d}; \mu, \sigma) = \prod_{i=1}^n \frac{1}{\sigma} \varphi(\delta_i - \mu). \quad (5.2.2)$$

The likelihood for n non-destructive measurement is a bit harder to derive. As also mentioned in Chapter 2 there are two models that can be applied to describe the measured non-destructive data.

²Note that in the previous subsection the random variable $H = \Sigma^2$ was used, using that model is the same up to the transformation, thus the same predictive distribution is found for R .

³Actually it is due to setting the normal model on the parameter $\Sigma(\sigma)$, the variance of the strength of the timber which makes it into a non-conjugate model.

⁴These three kinds are destructive data, censored destructive data, and non-destructive data.

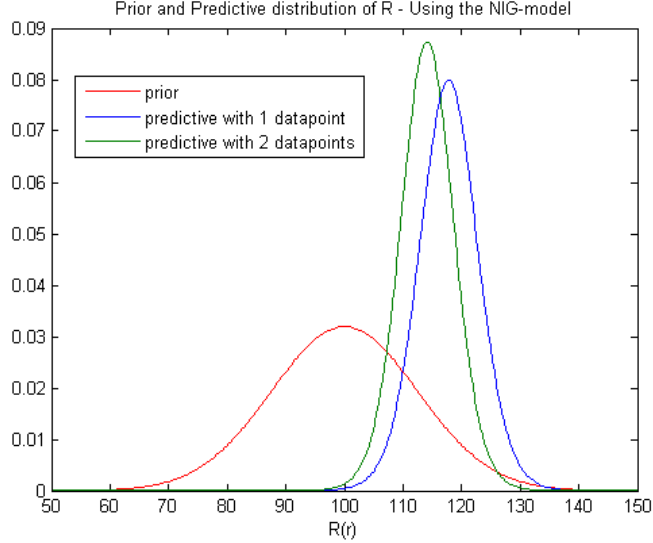


Figure 5.2: A plot of the prior and predictive distribution

In case of an additive error model (Equation 2.0.7) the likelihood is found by using the LTP in the following way,

$$\begin{aligned}
 L_{\underline{\Upsilon}}(\underline{v}) &= f_{\underline{\Upsilon}|M,\Sigma}(\underline{v}; \mu, \sigma) \\
 &= \int f_{\underline{\Upsilon}|M,\Sigma,\varepsilon}(\underline{v}; \mu, \sigma, a) f_{\varepsilon}(a) da \\
 &= \int \left[\prod_{i=1}^n f_{\Upsilon|M,\Sigma,\varepsilon}(v_i; \mu, \sigma, a) \right] f_{\varepsilon}(a) da \\
 &= \int \left[\prod_{i=1}^n f_{R|M,\Sigma}(v_i - a; \mu, \sigma) \right] f_{\varepsilon}(a) da.
 \end{aligned} \tag{5.2.3}$$

LTP works in this case as the measurements are again conditional independent given the error-term (ε). Note that the Jacobian from the transformation from Υ to R equals 1,

$$J = \left| \frac{\partial R}{\partial \Upsilon} \right| = 1 \tag{5.2.4}$$

In the direct numerical integration approach we first define the discretisation of the ε -domain as $\langle a_j \rangle_{j=1}^{p_a}$ and then the integral in Equation (5.2.3) is transformed to the following summation.

$$L_{\underline{\Upsilon}}(\underline{v}) = \sum_{j=1}^{p_a} \left[\prod_{i=1}^n f_{R|M,\Sigma}(v_i - a_j; \mu, \sigma) \right] f_{\varepsilon}(a_j) \Delta_j^{p_a} \tag{5.2.5}$$

The following variables were introduced; p_a , $\Delta_j^{p_a}$, the number of points used in the discretisation and the distance between discretisation-points, respectively. As all distributions in Equation (5.2.5) are

known⁵, this defines the likelihood in the direct numerical integration approach using the additive model.

Using the multiplicative model (Equation 2.0.8) to describe the non-destructive data results in a different likelihood,

$$\begin{aligned} L_{\underline{\Upsilon}}(\underline{v}) &= \int \left[\prod_{i=1}^n f_{\Upsilon|M,\Sigma,\varepsilon}(v_i; \mu, \sigma, a) \right] f_{\varepsilon}(a) da \\ &= \int \left[\prod_{i=1}^n f_{R|M,\Sigma}\left(\frac{v_i}{a}; \mu, \sigma\right) \frac{1}{a} \right] f_{\varepsilon}(a) da. \end{aligned} \quad (5.2.6)$$

The Jacobian of the transformation from Υ to R equals

$$J = \left| \frac{\partial R}{\partial \Upsilon} \right| = \frac{1}{\varepsilon}. \quad (5.2.7)$$

Applying the direct numerical integration approach to Equation 5.2.6 results in

$$L_{\underline{\Upsilon}}(\underline{v}) = \sum_{j=1}^{p_a} \left[\prod_{i=1}^n f_{R|M,\Sigma}\left(\frac{v_i}{a_j}; \mu, \sigma\right) \frac{1}{a_j} \right] f_{\varepsilon}(a_j) \Delta_j^{p_a}. \quad (5.2.8)$$

This concludes the discussion on the likelihood function.

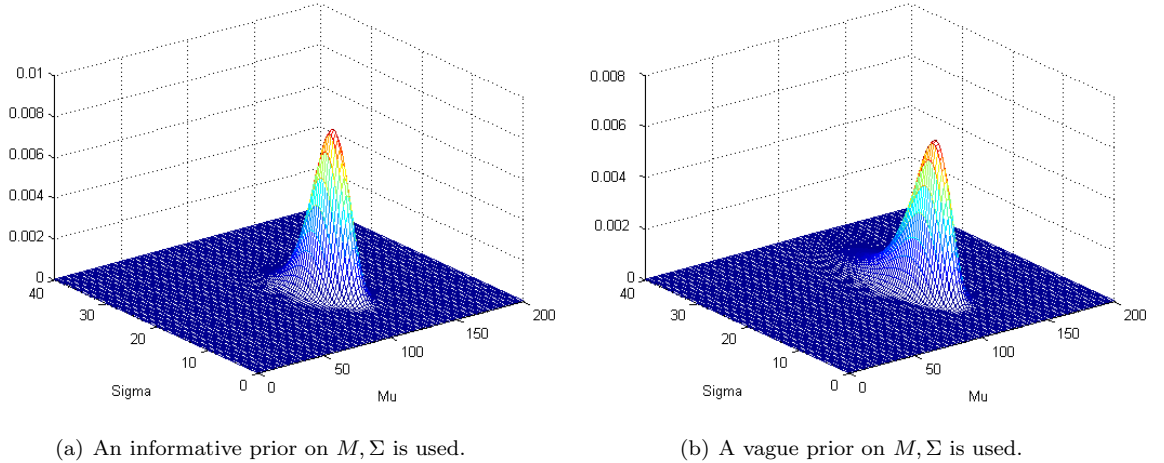


Figure 5.3: Posterior distributions for M, Σ using an informative and vague prior with the Camaru data-set

The update can now be defined by substituting the likelihood and the prior in Bayes Rule. This then gives the posterior distribution over (M, Σ) . Applying LTP results in the predictive distribution for R . A MATLAB script is added in the Appendix which is implemented using this approach, see Appendix C, and [10]. Figure 5.3(a) shows the result of using this m-file. Plot a in this figure has been created using an informative prior, and plot b using the vague prior. Notice that the posterior

⁵ $(R|M, \Sigma) \sim \mathcal{N}(M, \Sigma)$ and $\varepsilon \sim \mathcal{N}(0, \sigma_{\varepsilon})$.

(unresolved) uncertainty (reflected by Σ) is greater when using the vague prior than when using the informative prior. This is even better displayed when evaluating⁶ the plot with both calculated predictive distribution of R , see Figure 5.4.

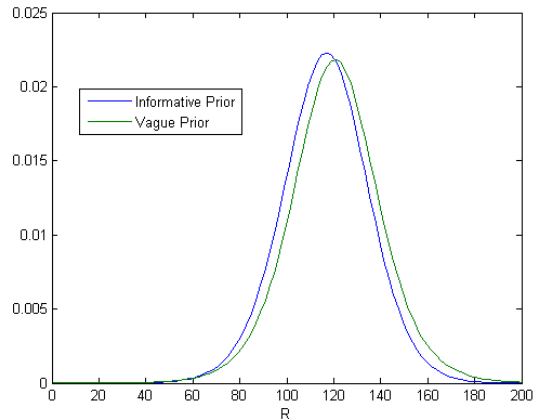


Figure 5.4: The predictive distribution of R using vague and an informative prior with the Camaru data-set

As was shown, the procedure for calculating the posterior distribution over $M(\mu)$ and $\Sigma(\sigma)$ is straight forward using the definition of Bayes' rule. When using non-destructive data the likelihood function becomes a bit more complicated because LTP had to be used. By applying LTP an extra dimension is added to the calculation. If LTP has to be performed many times multi-dimensional objects need to be handled in the direct numerical integration approach. This will increase the computational complexity, and thus will influence the computation time. Therefore this approach is very robust and in principle will generate an answer. However in practice application the computation time and memory needed to perform the calculation can become very large. This is the main drawback of using direct numerical integration, the discretisation that it uses can make the computation very large.

5.3 Application of the BBN Approach

The following model is created in HUGIN representing the timber case, see Figure 5.5. This HUGIN-model was created using the HUGIN API and the m-file that was used can be found in Appendix C. The graphical model used in HUGIN needs the following explanation. In Figure 5.5 three colored boxes have been drawn indicating: the forward model, destructive measurements, and non-destructive measurements. The forward model and destructive measurements are straight forward as they indicate a dependency on Mu and $Sigma$. The non-destructive measurements are a bit more complicated, thus let us discuss one non-destructive measurement. The node $ND1$ represents the strength of the (non-destructive) tested beam. This is not the strength that is measured, denoted by node ND_M_1 , as this is also influenced by the measurement error, node A . The nodes MuA and

⁶his effect is even more apparent when using lesser data to perform the update.

$Variance_A$ denote the fixed parameters of the normal distributed prior of node A . Measurements i (evidence) on a random selected beam are thus entered in the nodes ND_M_i . The quantitative model that was used in this BBN is straight forward entering the models discussed in Chapter 2. As discrete chance nodes are used a discretisation needs to be defined for these nodes. In this BBN the discretisation was found by trail and error, meaning that for different discretisations the network was ran and the results inspected until the results were proper and the detail was fine enough. The result can be seen in de m-file used to create this BBN.

The posterior distribution for R created in HUGIN is exported to MATLAB, and a plot is made of it together with the posterior derived by direct numerical integration. From this plot, Figure 5.6, it is clear that HUGIN gives the same result⁷.

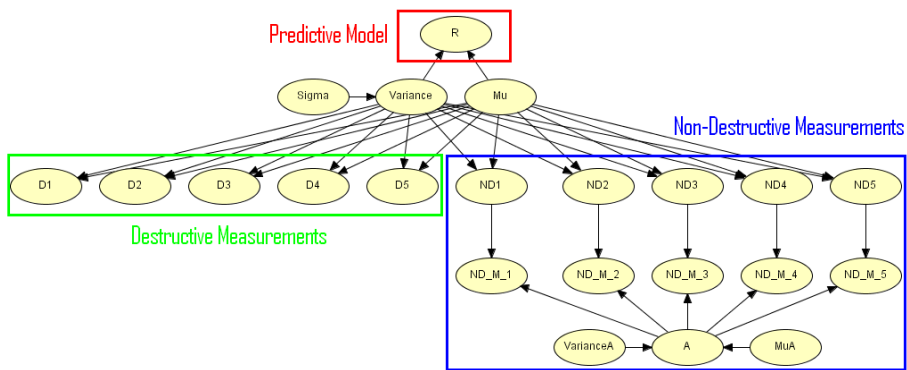


Figure 5.5: The graphical model from HUGIN used in the timber case

When comparing the computation time of both approaches it was seen that HUGIN is on average slower, depending mostly on the size of the discretisation. Do note that in HUGIN computing the posterior for different data using the same model is very fast. This is because HUGIN first compiles the model which takes the longest, and then propagates the evidence. Thus if new evidence (data) needs to be entered, the model no longer needs to be compiled. In direct numerical integration the entire model needs to be computed all over. Another benefit that using the BBN approach is it easier explains the used models this as it holds a visualization of the models.

⁷The small discrepancy there is, is due to discretisation used in HUGIN. This was checked by running a smaller network with a denser discretisation and then the two methods gave the same results.

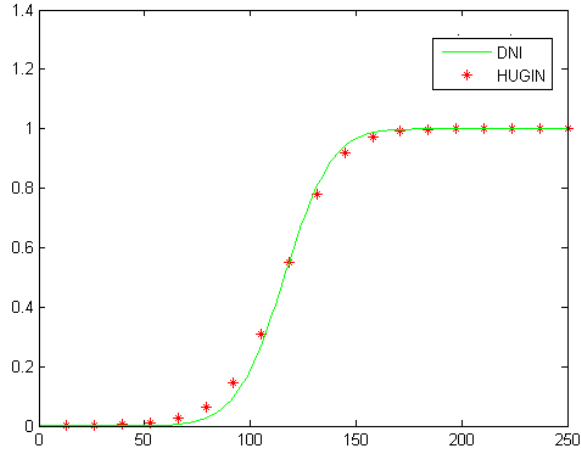


Figure 5.6: A plot of the posterior distribution derived by direct numerical integration and HUGIN

5.4 Application of the Bayes Linear Approach

After some application of the Bayes linear approach to various settings in the timber case, this approach only seemed applicable in certain specific settings. For example it performs well in the simplified timber case. Why this approach does not seem to work proper in other cases was investigated by making comparisons between Bayes linear and direct numerical integration and the analytical approach. This will show that Bayes linear is not applicable in these general settings as it only captures the first moment and fairly underestimates the second moment.

First a comparison is made between Bayes linear and direct numerical integration. All the calculations needed for Bayes linear using the \mathcal{N} -model in the timber case are added as proofs in the Appendix A.2. The following table, Table 5.4, shows the first two moments of R determined by Bayes linear and direct numerical integration using the \mathcal{N} -model on the basis of destructive data of Camaru data-set.

Table 5.4: First two moments of R determined with Bayes linear and direct numerical integration

Used Approach	Mean	Variance
Bayes linear	116.3058	332.4596
Direct numerical integration	115.1472	439.6726
Difference	1.1586	-107.2130

From Table 5.4 it is concluded that Bayes linear and direct numerical integration using the \mathcal{N} -model do not give the same results. It is seen that Bayes linear fairly underestimates the posterior variance. To investigate this even further the following comparison is made between Bayes linear and the analytical approach in the form of a conjugate analysis using the \mathcal{NIG} -model.

Bayes linear when using the \mathcal{NIG} -model is compared to the analytical approach. To perform Linear Bayes the following properties are needed; $\mathbb{E}(R)$, $\mathbb{E}(D)$, $\mathbb{V}ar(R)$, $\mathbb{V}ar(D)$, and $\mathbb{C}ov(R, D)$.

These are all stated by the following five propositions, the calculations are posted as proofs in the Appendix A.2.

Proposition 5.4.1. *The expectation of the strength of the timber(R) equals m ; i.e.*

$$\mathbb{E}(R) = m \quad (5.4.1)$$

Proposition 5.4.2. *The expectation of a destructive test equals that of the strength of the timber, i.e.*

$$\mathbb{E}(D) = \mathbb{E}(R) \quad (5.4.2)$$

Proposition 5.4.3. *The Variance of the strength of the timber is given by*

$$\text{Var}(R) = \frac{1 + \Lambda}{\beta(\alpha - 1)} \quad (5.4.3)$$

Proposition 5.4.4. *The variance of a destructive test result is the same as that of the strength of timber.*

$$\text{Var}(D) = \text{Var}(R) \quad (5.4.4)$$

Proposition 5.4.5. *The Covariance between the strength of the timber and a results of a destructive test is*

$$\text{Cov}(R, D) = \frac{\Lambda}{\beta(\alpha - 1)}. \quad (5.4.5)$$

According to analytical calculations using the \mathcal{NIG} -model the expectation of the predictive distribution of R equals;

$$E_D = \frac{m + \Lambda \sum D_i}{1 + n\Lambda}, \quad (5.4.6)$$

which can also be written as

$$\begin{aligned} E_D &= m + \frac{\Lambda \sum D_i - mn\Lambda}{1 + n\Lambda} \\ &= m + \frac{\Lambda \sum (D_i - m)}{1 + n\Lambda} \\ &= m + \frac{\Lambda}{1 + n\Lambda} 1_{n,n} [\underline{D} - \underline{m}]. \end{aligned}$$

In one dimension this equation reads

$$E_D = m + \frac{\Lambda}{1 + \Lambda} (D - m). \quad (5.4.7)$$

In this form a clear comparison between Bayes linear and the analytical solution can be made. The first moment is exactly the same for both methods. Therefore in the one dimensional case Bayes linear approach is exactly the same as the analytical approach.

The variance of the posterior distribution of R determined by the analytical approach is given as

$$\text{Var}_D(R) = \frac{1 + \Lambda}{\beta(\alpha - 1)}. \quad (5.4.8)$$

In one dimension the second moment of R is described by Bayes linear as

$$\begin{aligned} \text{Var}_D(R) &= \text{Var}(R) - \frac{\text{Cov}(R, D)^2}{\text{Var}(D)} \\ &= \frac{1 + \Lambda}{\beta(\alpha - 1)} - \frac{\Lambda^2}{\beta^2(\alpha - 1)^2} \frac{\beta(\alpha - 1)}{1 + \Lambda} \\ &= \frac{1 + \Lambda}{\beta(\alpha - 1)} - \frac{\Lambda^2}{\beta(\alpha - 1)} \frac{1}{1 + \Lambda} \\ &= \frac{1 + 2\Lambda}{\beta(\alpha - 1)} \frac{1}{1 + \Lambda}. \end{aligned} \quad (5.4.9)$$

Equations (5.4.9) and 5.4.8 are not equal, this shows that the second moment of Bayes linear is not the same as the one derived by an analytical approach. Therefore using the second moment of Bayes linear seems not useable. Notice that the difference between Equation (5.4.8) and (5.4.9) is that the parameters α and β are not adjusted, which means no update of $H(\eta)$ is performed using Bayes linear.

This section has shown that in general Bayes linear approach does not give equal results as the direct numerical and analytical approach. Therefore Bayes linear will not be further discussed in this thesis as more research is needed to explain the differences in results. Still it is noted that the assumptions (prior information specification) needed to perform the Bayes linear approach are less strict then defining a full prior distribution needed in the other approaches. In cases where this might not be possible Bayes linear can possibly be used.

Chapter 6

Decision Problem

6.1 Introduction to the Decision Problem

In the previous chapter the interpretation of observations was discussed. It gave four different approaches to apply the Bayesian method. It also showed the effect of using an observation. From this it was noted that the effect of one observation can be very different from a second or another observation. In a reaction to this varying result of using observations the following questions are raised. How should observations be performed, or even should an observation be performed at all, and if so when or how many need there be performed. This chapter will try to answer some of these questions. The main focus here is on the number of tests that need to be performed.

In order to compare different strategies for observations, their consequences, and the decisions of performing such a strategy a measure needs to be defined. For this the utility defined by Savage [14] is used. The complete theory used here for the decision making is called Savages decision theory. A concise introduction of it is given in the following section. The following sections will give the consideration defining as well as the approaches defined to solve the decision problem. Also a section is devoted to the application of this theory in the timber case.

6.2 Savage's Decision Theory

In order to discuss what is called Savages decision theory (SDT)[14] first Savages utility theory needs to be discussed, which is the main component of SDT. This section contains only necessary mathematical proofs and details, that are needed to formally discuss this theory. Only the main ingredients and the use of the theory will be shown here. For more details see [14], [20], and [17].

Savage's approach to utility is based on preference relations over acts. This seemingly easy concept forms the basis of his entire theory. Savage introduced the concept that preference (of a single rational decision maker) among different acts is made on the basis of expected utility. This was done by distinguishing two components defining a preference relation over acts, namely a (quantitative) probability and a measure of the acts/consequences called utility. This unique probability represents the uncertainty in a possible outcome of an act. The utility function rates the outcomes, such that the expected utility reflects the preference relation. This can be expressed mathematically with a theorem, therefore first define the following.

The non-empty set \mathcal{S} is the set of all states of the world. \mathcal{C} is the finite¹ set containing all consequences which are the possible states of the acting subject, i.e. outcomes of the act. \mathcal{F} is the set of all acts i.e. $\mathcal{F} = \{f | f : \mathcal{S} \rightarrow \mathcal{C}\}$. Then also define a utility function as $U : \mathcal{C} \rightarrow \mathbb{R}$.

Theorem 6.2.1 (Preference and Utility). *Let f and g be two acts, act f is preferred to act g if and only if $v(f) \geq v(g)$. Where $v(f)$ denotes the expected utility,*

$$v(f) = \sum_{c \in \mathcal{C}} p(f^{-1}(c))U(c) \quad (6.2.1)$$

Proof. A very nice proof of this theorem is given in [20]. (It is not given here as it needs much more mathematical detail and rigor in the definitions.) □

Remark 6.2.1. Note that the expected utility of an act can also be defined as

$$v(f) = \sum_{s \in \mathcal{S}} U(f(s))p(s). \quad (6.2.2)$$

As the theorem and definitions given above are very formal, an interpretation is given by stating an example after the concept of a decision tree is given.

A decision tree is an event tree which graphically represent the decision process (see the example below for an example of a decision tree in Figure 6.2). The notation that is used in this report is generally adopted from the notation used by Ada Decision Systems, see [6]. In the notation square nodes represent the decisions and circular nodes uncertain variables/ events. The expected utility and maximal (expected) utility is displayed between squared bracket above each chance and decision node respectively. Above the end nodes the utility for that state² is displayed. The rules for evaluating a decision tree are fairly simple as for each end node the utility is calculated and then the tree is 'rolled-back' by taking the expectation over all states for chance nodes and maximizing the expected utility for each decision node.

Example 6.2.1. *A concert promotor wants to hold a jazz concert in Zoetermeer. This promotor has to make a decision which location he will use for this concert. After some research the promotor found two locations fit for his concert. These two possible locations are indicated as {Indoor, Outdoor}, as this characterizes these locations. From previous concerts the promotor knows that the weather influences ticket sales. Table 6.1 shows this influence on the expected profit of the jazz concert.*

Table 6.1: Expected profit of the jazz concert

Weather / Location	Outdoor	Indoor	Probability
Sunshine	100	50	0.33
Cloudy	20	60	0.67

¹This is a technical requirement to ensure that a decision problem is well defined.

²Each end node represents a state of the world which can be evaluated by the utility function.

Note that this table shows that if the promotor is certain over the weather he will hold the concert indoor if it is cloudy and outdoor if it is sunny. As the weather is uncertain probabilities are assigned, displaying that it is cloudy more often than the sun shines. If the promotor should decide on this information he act to hold the concert indoor as is displayed by the decision tree in Figure 6.1.

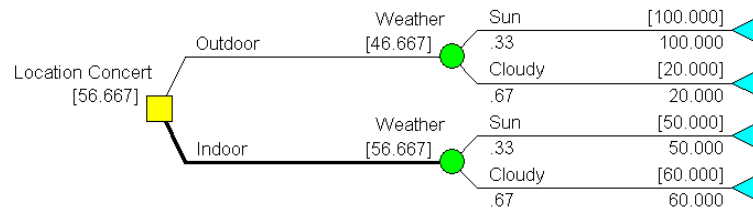


Figure 6.1: The optimal decision for the location of the concert displayed in a decision tree, where the (maximal) expected profit is displayed in the squared brackets

The promotor now has the possibility to buy information from a weather bureau who will perform a test which will indicate the weather. The promotor now asks himself if this information is worth its price, or should he not buy this information. The following information is available on this test. The test has such an accuracy that it gives an accurate prediction in 9 out of 10 cases³. Combining this with the promoters own information means that the following probabilities can be calculated. Note that the first entry in Table 6.2, '0.82' denotes $P(\text{Weather} = \text{Sunshine} | \text{TestResult} = \text{Good})$.

Table 6.2: Information on the weather-test

Probability of — Test Result	Good	Bad
Sunshine	0.82	0.05
Cloudy	0.18	0.95
Probability	0.37	0.63

The decision tree in Figure 6.2 shows the optimal decision(s) for the promotor. Note that when a test is not performed no information is gained from it, thus the lower half of the decision tree is symmetrical as is shown in the decision tree from Figure 6.1. Also note that the optimal decision for the location of the concert changes under the influence of the test. The optimal decision for the location is if no test is performed or the test result is 'Bad' to decide for the location to be 'Indoor', and if the test result is 'Good' to be 'Outdoor'. In the decision tree displayed by Figure

³This means $P(\text{Good} | \text{Sunshine}) = P(\text{Bad} | \text{Cloudy}) = 0.9$

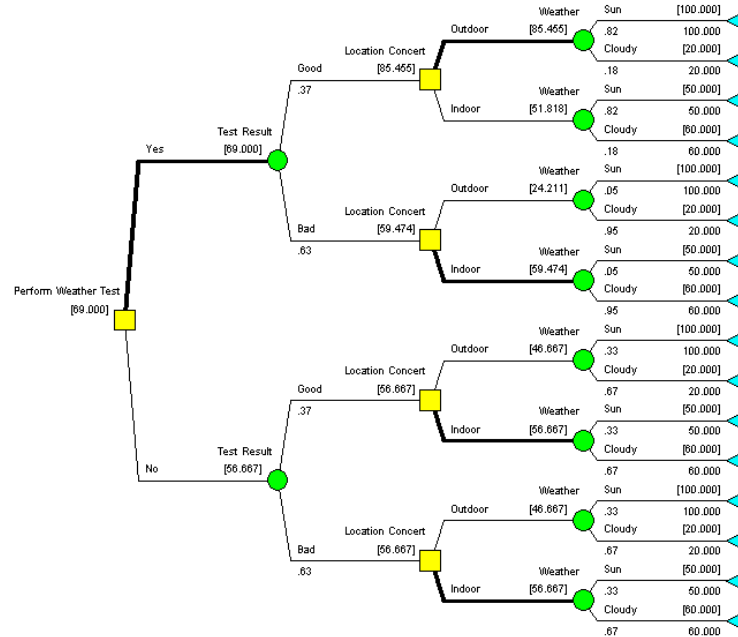


Figure 6.2: The decision problem to perform a test to decide the location of the concert displayed in a decision tree

6.2 no cost of the test were taken into account. Still it can be seen that if the cost of testing is below $12.333 (= 69 - 56.667)$ it should be performed. This number (12.333) is called the value of information of the Test⁴.

In the previous example SDT was applied. Notice it uses the reverse reasoning from the utility theory. Where as the utility theory starts from reasoning from preference relations to define utility, and decision theory starts by calculating the utilities and then infers the decision that needs to be made, thus displaying the preference relation.

The following remark needs to be made when using SDT.

Remark 6.2.2. The optimal decision need not be unique.

Note that optimal means the decision with the highest expected utility. It is possible that some decision have the same expected utility, therefore making the decision maker to be indifferent over the two alternatives. When dealing with the decision problem concerning testing and there exist two strategies that lead to the optimal decision, then the arbitrary choice is made to define that strategy using the lowest number of tests as the (unique) optimal. This as it uses less capacity for testing.

⁴ Value of information is formally discussed in the next chapter.

Chapter 7

Approaches to the Decision Problem

7.1 Considerations Decision Problem

The questions that define the decision problem concerning observations can be categorized into three groups. The questions in each category all focus on a different issue of the decisions problem. These three categories are

- Observation types,
- Number of observations,
- Time of observation.

The first category concentrates on different types of observations, because different type of sensors can be used to perform the observations, for instance destructive and non-destructive tests. The second category concentrates on the number of observations that are performed using one type of test. The third category concentrates on when to observe. In general observations are done when the reliability drops below the pre-defined requirement or a fixed inspection interval¹ is used.

As mentioned in the previous chapter, the approaches defined in this chapter concentrate on finding the optimal number of observations that need to be performed, the second category in the above list. As decisions need to be made on the basis of observations and the thereby implied (predictive) posterior before they are known, this problem can be classified as a pre-posterior analysis.

In order to find the optimal number of tests, solving the decision problem, an utility function needs to be defined. This utility function will measure the effects of a decision. This utility function is assumed to be additive, such that it is separable into two parts. The first part being dependent on

¹A fixed inspection interval can be to inspect every 10 years.

the test outcome (the result of the decision) called the gain. The second part not being dependent on the decision result, i.e., the cost of testing².

A typical behavior of the expected utility compared to the number of performed test is that displayed by curve three in Figure 7.1. This typical behavior can be explained by individually examining the behavior of the two parts of the utility function. The expected gain (first part of the utility function), displayed as curve one in Figure 7.1. The expected gain clearly shows three stages of behavior, these can be explained by the effect of a test on the posterior distribution. The first stage shows a more or less constant behavior. The first tests usually will not have that much effect on the posterior distribution as the prior information will dominate the posterior distribution. Therefore the expected gain will not change much under the influence of more tests. The second stage in the behavior of the expected gain which is increasing can be explained by the following. As more tests are performed the prior information no longer dominates the posterior and the effect of performing a test increase which explains the (local) increasing behavior of the expected gain. The third stage shows that for a sufficient large number of tests the expected gain approaches a limiting value. When enough data is used the posterior will converge to the sampling distribution of the data, thus the posterior will not change much under more tests, explaining the constant behavior.

The behavior of the second part of the expected utility (cost of testing) is shown by curve two in Figure 7.1. This behavior is the result of the (in this case) assumption of proportional character of costs of testing in terms of to the number of performed tests.

Concluding the typical behavior of expected utility, which equals expected gain minus the costs of testing, is displayed by curve three. This curve clearly shows an optimal number of tests at \tilde{n} . The number of test that can be performed is an integer number, thus the optimal number of tests is defined as the integer closest integer to \tilde{n} .

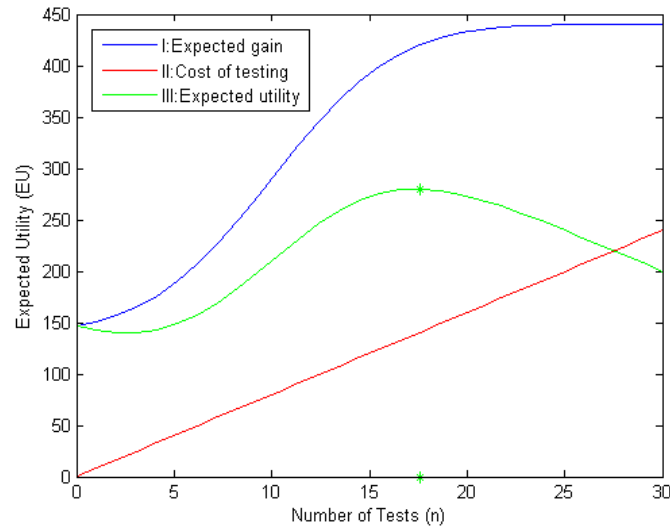


Figure 7.1: A plot of the expected utility and costs against the number of performed tests

²The cost of performing a test are only dependent on the decision as follows; they exist when the test is performed and are zero if the test is not performed. This in contrary to the gain, the gain can be positive if the test result is favorable and negative if the test result is bad.

The general method for solving these decisions problems is two folded, first the decision problem (utility function) needs to be defined, which involves defining a measure for the gain and cost of testing. Second on the basis of this utility function infer the optimal decision, thus the calculations need to be performed. From a theoretical point of view this is very clear and sound. The only restriction is that the space of consequences (test results) must be finite, which is not the case theoretically as the distribution over the observation is continuous. In practise this method can be hard to follow, this as the computational complexity can become very large. The approaches for implementing each of two steps in this method are discussed separate in the following two sections, after which they are applied to the timber case.

7.2 Defining the Decision Problem

In order to measure the effects of performing an observation an utility function needs to be defined, the following two different approaches are defined to perform this task,

- Design approach,
- Producers approach.

These two approaches differ in their interpretation that they use to define the utility function. The design approach defines them from a user perspective with cost of failure, maintenance and building costs under consideration. Opposed to the approach from a producer perspective which uses sales revenues as main considerations. This aspect of the decision problem is hard to formulate for a general setting other then given above. In Section 7.4 which discusses an application of the decision problem both approaches are discussed.

7.3 Approaches Solving Decision Problem

Given that the utility function is defined the decision problem is formulated and it needs to be solved. Three different approaches to solve the decision problem have been considered during the research of this thesis. They are formally stated here and will be applied to the timber case in the next section.

- Decision Tree
- Fixed Size Test Plan
- Value of Information

The first approach is computing/calculating the decision tree, this was demonstrated in Example 6.2.1. Calculating a decision tree can sometimes become impossible as the computational complexity becomes too large. The computational complexity of the decision tree is determined by two factors the number of tests that are considered and the discretisation of the chance nodes (test results) in the decision tree. The computational complexity of a decision tree for a typical testing situation is given by the following theorem.

Theorem 7.3.1. *The computational complexity of a decision tree deciding on whether to perform at most $n \geq 0$ tests and using a discretisation of m possible test outcomes is defined as:*

$$\text{Complexity} = \mathcal{O}(m^{n-1}). \quad (7.3.1)$$

Proof. The number of states that need to be evaluated equals

$$\begin{aligned} \sum_{i=0}^n (m)^{i+1} &= \sum_{i=-1}^{n-1} (m)^i \\ &= \frac{(m)^{n-1+1} - (m)^{-1}}{m-1} \\ &= \frac{m^{n+1} - 1}{m(m-1)}. \end{aligned} \quad (7.3.2)$$

This as when no test are performed only m states of the predictive distribution need to be evaluated, e.g., evaluating the prior information. If at most 1 test is evaluated all tests of performing at most none tests need to be evaluated as for each tests outcome (m) the predictive distribution needs to be evaluated in m states which means $m + m^2$ evaluations. Thus for at most n test Equation (7.3.2) is found. \square

This means that when $n = 11$ tests are considered and a discretisation of $m = 8$ points is used the complexity is $8.6 \cdot 10^8$ paths need to be evaluated. This makes this method only applicable for small number of tests with small discretisations. As the decision tree is a great visualization of the decision process it can be used for visualization with minor alterations. This is shown in subsection 7.4.2.

The second approach is defined as the fixed size test plan. This approach can also use a loss function, which is (often) the negative of the utility function (including the costs of performing a test). With the help of a loss function the expected loss is calculated for each number of tests also called the Bayes Risk. Then the minimum of this loss function is determined with respect to the number of tests. By definition this minimum is the optimal number of tests. This fixed size test plan has also been discussed³ in [13] and [4]. This approach is formally represented by the following definitions and theorem⁴.

Definition 7.3.1 (Bayes decision rule). A *Bayes decision rule* (\tilde{r}) is a decision (and act) which minimizes the posterior expected risk.

Definition 7.3.2 (Bayes risk). Assume that $L^F(r, d, n)$ is the loss in observing D^n and taking decision d . Subsequently assume that R has the prior (predictive) density $f_R(r)$. Let \tilde{r}_n denote the

³The fixed test plan is discussed in these books as determining of sample size.

⁴These definitions and theorem are rephrased version of the ones given in [13].

Bayes decision rule for this problem (if one exists), then

$$BR(n) = \mathbb{E}_R \mathbb{E}_{\underline{D}^n | R} [L^F(r, \tilde{r}_n, n)] \quad (7.3.3)$$

denotes the Bayes risk. And let \tilde{r}_0 and $BR(0)$ are the corresponding quantities for the no-observation problem.)

Theorem 7.3.2 (The optimal fixed size test plan). *The optimal fixed size test plan is clearly that n (with $n \geq 0$ and integer) which minimizes $BR(n)$.*

The third approach is the value of information (VOI) discussed in [17]. This approach can be viewed as a formalization of the approach defined by the decision tree, see Example 6.2.1. This approach calculates the VOI of a test and from it derives the optimal decision. This approach can be viewed as determining the derivative of the curve created in the fixed size test plan approach. The VOI of a test equals the (maximal) information (in terms of utility) that a test is expected to give. Thus by determining for each test when this becomes negative defines the optimal test size. The following definitions are given to define the VOI and the next remark explains when to observe.

Definition 7.3.3 (Expected value of a set of acts). The expected value⁵ of a set of acts is defined as

$$\forall f \in F \text{ the value of } F \text{ is } v(F) = \max_{f \in F} v(f) \quad (7.3.4)$$

Definition 7.3.4 (Conditional expected value of a set of acts). The conditional expected value of a set of acts is therefore defined as;

Given a test(T) with possible values $t \in \tau$, and $\forall f \in F$ the expected value of F with respect to $p(\bullet | T = t)$ is

$$v(F|T = t) = \max_{f \in F} v(f|T = t) \quad (7.3.5)$$

Definition 7.3.5. The *value of information* (VOI) of a test(T) is defined as follows

$$v(T) = v(F|T) - v(F). \quad (7.3.6)$$

Remark 7.3.1. It is only beneficial to observe (perform a test) if the value of information of a test is greater than the costs of observation.

The following theorem also taken from [17] shows that if tests are free as much as possible test should be done.

Theorem 7.3.3 (Increasing value of information). *For any random variable T and any $f \in F$, $v(F|T) \geq v(F)$.*

⁵The expected utility for a single act, $v(f)$ is defined in Theorem 6.2.1.

Proof.

$$\begin{aligned}
 v(F|T) &= \sum_{t \in \tau} \max_{f \in F} v(f|t)p(t) \\
 &\geq \sum_{t \in \tau} v(f|t)p(t) \\
 &= v(f)
 \end{aligned}$$

□

The approach followed when using VOI is calculating the VOI of performing n tests for different n and then determining when it becomes constant. This approach will be shown in the next section.

The above given three approaches do not differ that much from each other. The decision tree can be used to visualize the two other approaches as both can be derived from the decision tree. The decision tree can also be used to solve more complex decision problems as the sequential decision problem. This can also be done with the VOI-approach.

7.4 Application to Timber Case

7.4.1 Considerations Timber Case

The timber case is going to be used to demonstrate how to solve the decision problem concerning observations. Destructive testing has been chosen as the test type under consideration. This choice has been made as the interpreting⁶ of the observations using destructive measurements in the timber case can be done analytically (conjugate analysis).

The ultimate goal is of course determining the strength of the population of beams under investigation (batch). The decision problem defined in the timber case is how many tests should be done to determine this population strength(characteristic strength). The primary variable in the timber case is R , all decision will be related to the (predictive) distribution of R . If testing was free as many tests as possible would be done, this is not the case. The effect of a test needs to be compared to the cost of performing a test. As the effect of an test is uncertain the expected effect is used. An utility function needs to be defined to measure the effect when having a test result. The following interpretations of the utility function were considered in the timber case.

As a first attempt this utility function was defined as

$$U(\delta_1, \dots, \delta_k) = C_1 \mathbb{E}(R|\delta_1, \dots, \delta_k) - C_2 \sqrt{\text{Var}(R|\delta_1, \dots, \delta_k)} + C_3 - kC_T, \quad (7.4.1)$$

where the constants C_1, C_2 , and C_3 are chosen appropriate such that the utility function reflects the preference of the decision maker (and the dimensions are correct). The cost of testing C_T are assumed to be proportional to the number of tests. This utility seemed very natural as maximizing it implies increasing the mean and lowering the variance. This utility function was used to explore the possibilities of the decision tree. The natural choice of $C_1 = 1$, $C_2 = 1.6449$, $C_3 = -\mathbb{E}(R) + \text{Var}(R)$ was taken such that the utility reflected the characteristic value of the timber. This inferred the difficulty of defining C_T appropriate, in the following examples it was taken as 3.

⁶See Chapter 3, the Bayesian method, on page 9.

The use of this utility function was abandoned and the problem was approached as a parameter estimation problem⁷. This meant that a loss function was defined which can be seen as the negative of the utility function. This as maximizing expected utility equals minimizing the expected loss. Still an optimal number of test is to be found by balancing the decision loss and the cost of testing. The following two loss functions were used.

$$\mathcal{L}(\tilde{r}, r) = K(\tilde{r} - r)^2 + C(n) \quad (7.4.2)$$

$$\mathcal{L}(\tilde{r}, r) = \begin{cases} c_1(r - \tilde{r}) + C(n) & r \geq \tilde{r} \\ c_2(\tilde{r} - r) + C(n) & r \leq \tilde{r} \end{cases} \quad (7.4.3)$$

The following considerations were used in the timber case to define the constants needed in these loss functions.

Design approach

The design approach uses the perspective from a design point of view to define the loss or utility function. This means that the following considerations are used. When achieving a higher(lower) characteristic value for the tested material implies a reduction (addition) in materials needed to be used in a construction (while retaining reliability). Or when testing is done on existing materials a better(worse) situation is found an increased (decreased) reliability occurs which means longer(shorter) life time or less(more) repairs. Calculating these effect for a standard situation can define the constants by taking for instance average achieved savings and average expected costs of failure.

Producers approach

From a producers perspective an entirely different interpretation can be given to the decision process. The producer wants to sell the tested batch of materials, they are a-priori graded by the characteristic strength in a strength class. As the tests can indicate that the material he wants to sell is stronger or weaker a clear indication to the loss/utility function can be given. A small deviation in the prior graded strength compared to the posterior is of no interest to the producer unless this means that the material needs to be graded in a different characteristic strength class. In that case the material can be sold for a higher or lower price which will increase or decrease the revenue of the batch of material or even make it unsalable.

7.4.2 Application Decision Tree

This subsection will show the application of the *decision tree* in the simplified timber case⁸. It was already noted that this approach is only suited for small decision problems. The utility function given by Equation (7.4.1) is used. The decision tree in Figure 7.2 is created by DPL in case of at most two tests can be performed and a discretisation of three intervals is used. The probabilities shown in this decision tree are calculated using Table 7.1. This table shows the parameters used in the normal distribution that the define the test results in the simplified timber case. Note that the test result of the first test is entered in the parameter of the probability of the second test result.

⁷Note that this approach sees R as the parameter, although the update is done by updating the parameters M, Σ .

⁸The simplified timber case is used for its analytical expression for the predictive distribution of R.

The parameters in this table have been calculated using a conjugate analysis as was displayed in Section 3. The calculations performed by DPL to create this decision tree are applying Theorem 6.2.1. The decision tree shows that the optimal number of test in this case is to perform one test.

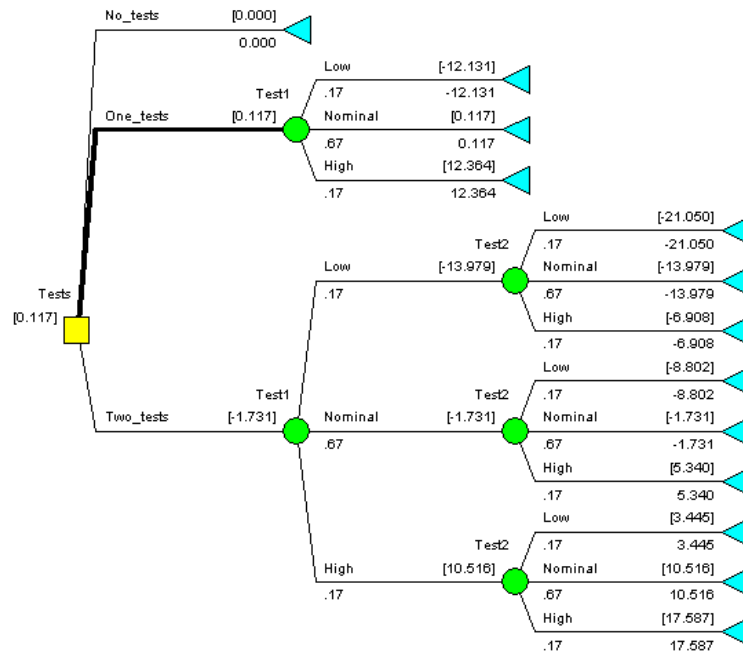


Figure 7.2: The decision tree in the timber case for at most two test and a discretisation of three intervals

The main reason for displaying this decision tree is not to show the calculations performed but to show the problems that arise when using a decision tree. As can be seen this tree becomes intractable for large number of tests and discretisations, therefore the following notation is introduced. A continuous chance node is a circle node with an circular arc on the outgoing arc. This outgoing arc represent all possible outcomes and the expected utility is presented above the node. The decision tree given in Figure 7.2 now becomes the one displayed in Figure 7.3(a).

Table 7.1: Parameters for the normal distributions used for all nodes in the decision tree

Node	Mean	Std. Deviation
Test1	100	$\sqrt{200}$
Test2	$\frac{100T_1 + 10000}{200}$	$\sqrt{50 + 100}$

The decision tree displayed in Figure 7.3(b) does not show all the information that the decision tree in Figure 7.2 did. It does however show the optimal decision, in this case performing one test, and the decision alternatives considered. Note that the continuous decision tree is compacter, and thus still useable as a visualization tool of the decision problem and performed calculations. In order to perform the calculations in the continuous version of the decision tree sampling is used by DPL.

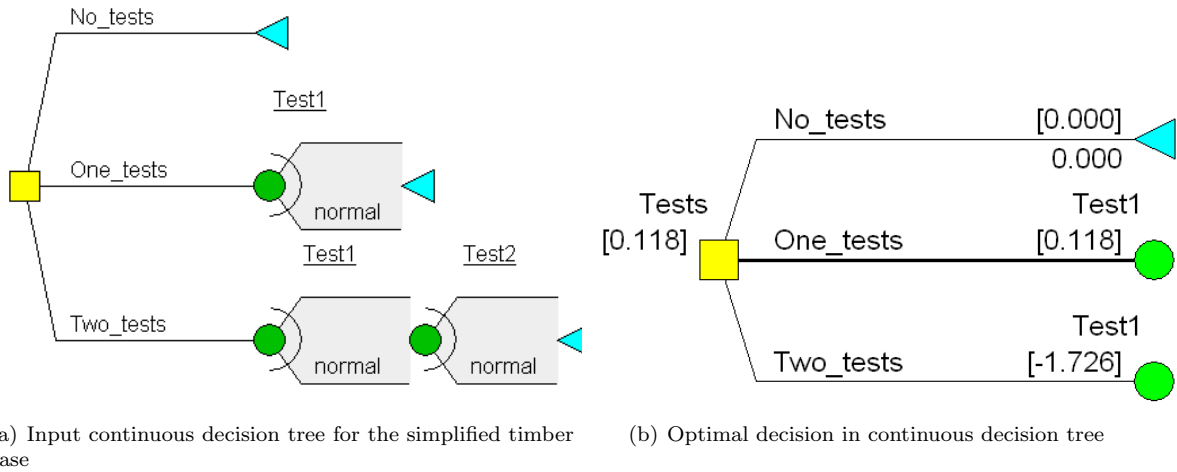


Figure 7.3: A continuous version of the decision tree in the simplified timber case evaluating at most two tests

In the next subsection an approach is shown how these calculations can be done by the ‘Fixed Size Test Plan’ approach.

7.4.3 Application Fixed Size Test Plan

This subsection will discuss how the optimal fixed size test plan can be determined for destructive testing in the (simplified) timber case. The optimal number of tests that is calculated is optimal with respect to the prior information. Therefore it is called ‘fixed size’ as it does not take into account the information gained from a test result other than was expected with respect to the prior information.

As first a quadratic loss function will be used, which will lead to an analytical expression for the curve displaying expected loss as a function of n . As second a bi-linear loss function is used. As third an in practice useable application is given.

Quadratic loss

The following loss function is used⁹

$$L^F(r, d(n), n) = L(r, d(n)) + C(n), \quad (7.4.4)$$

where the decision loss function is taken as $L(r, d(n)) = K(d(n) - r)^2$. Note that this (*quadratic decision loss*) function displays the loss in taking the decision to estimate R by $d(n)$. Note that in the timber case not the inference problem needs to be solved, the optimal number of tests needs to be solved. Thus the decision is made on n which implies a value for d . The quadratic loss function will be used to calculate the minimal posterior expected loss for a fixed number of tests (n), thus the decision is made to perform n tests. This is by definition the Bayes Risk. This risk (expected loss) is minimized over all n and the optimal fixed size test plan is found. Formally this approach

⁹Note that the super index indicates that it is the total loss for the fixed size test plan, thus including test costs.

is defined by Theorem 7.3.2. Applying this theorem in the simplified timber case (see Chapter 2) gives the following.

As quadratic loss is used the Bayes decision rule equals the posterior expectation¹⁰. This means that the Bayes risk in this case equals

$$\begin{aligned}
 BR(n) &= \mathbb{E}_R \mathbb{E}_{\underline{D}^n | R} [L^F(r, \mathbb{E}(R | \underline{D}^n), n)] \\
 &= \mathbb{E}_{\underline{D}^n} \text{Var}(R | \underline{D}^n) + C(n) \\
 &= K \frac{\sigma^2 \sigma_\mu^2}{\sigma^2 + n \sigma_\mu^2} + C(n).
 \end{aligned} \tag{7.4.5}$$

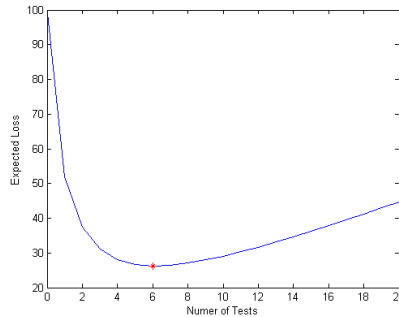


Figure 7.4: The decision loss against the number of test in the timber case

Again the cost of testing are taken proportional to the number of performed tests, i.e., $C(n) = nC_T$. A plot of the curve defined by Equation (7.4.5) is displayed in Figure 7.4. From this figure a clear minimum can be derived. This can also be done exact by taking n to be continuous and setting the derivative of $BR(n)$ to 0 ($\frac{\partial BR(n)}{\partial n} = 0$), which gives

$$\tilde{n} = \frac{\sigma \sqrt{K}}{\sqrt{C_T}} - \frac{\sigma^2}{\sigma_\mu^2}. \tag{7.4.6}$$

Now the optimal fixed size test plan is that non-negative integer n closest to \tilde{n} .

The followed approach gave the optimal fixed size test plan in the simplified timber case due to the following three reasons. As first, due to the use of a quadratic loss function a Bayes decision rule is easy found as the posterior expectation. Therefore the Bayes Risk is defined as the posterior variance. The second reason is the existence of an analytical predictive distribution. This implies that an analytical expression can be found for the posterior distribution. Together with the first reason gives an analytical expression for the expected loss, in this case the variance of the posterior distribution. The third reason is that the posterior loss is independent of the test results, only the number of tests performed. This implies that the posterior expected loss is still equal to the posterior variance. All of these reasons together give that an analytical expression is found for the optimal fixed test size.

It is very nice that an analytical expression is found for the fixed size test plan in the simplified timber case, still this does not represent a very realistic situation for two reasons. First the quadratic

¹⁰The proof of this is given in Appendix A.4.

loss function is symmetrical, which implies that overestimating the strength of the timber is equally penalized as underestimating the strength. As can be imagined this is not very realistic from the design and producers perspective. For instance from a design perspective clearly overestimating will lead to more (possible) failures and repair costs, which compared to the costs of underestimation will be greater and thus not symmetrical. Second the simplified timber case is not very realistic, and was only used for its analytical predictive distribution. Therefore another loss function is investigated.

Bi-linear loss function

A partial linear (non-symmetric) loss function will be investigated. The *bi-linear loss function* is defined below for inferring $d(n)$ by deciding to perform n tests and the true strength(R) equals r .

Definition 7.4.1. The bi-linear loss function is defined as follows

$$\mathcal{L}(d(n), r) = \begin{cases} c_1(r - d(n)) & r \geq d \\ c_2(d(n) - r) & r \leq d \end{cases} \quad (7.4.7)$$

where $c_1, c_2 \geq 0$.

Typically in determination of the strength of a material in the context of reliability $c_2 > c_1$. This was already mentioned from a designers perspective as overestimation is heavier penalized than underestimating. When using the bi-linear loss function the Bayes decision rule becomes $100 \frac{c_1}{c_1+c_2}$ -percentile estimate of the predictive distribution of R , which is proven in Appendix A.4. As the Bayes decision rule is the $100 \frac{c_1}{c_1+c_2}$ -percentile estimate of R an interesting interpretation to c_1 and c_2 can be given. The reasoning is as follows, as it is common to take the 5-percentile of the predictive distribution of R as the characteristic strength a condition on the ratio of c_1/c_2 is put as 1/19. This ratio is very helpful when defining the decision problem. Defining the cost of overestimating can often be done, but defining the cost of underestimation are much harder. By the defined ratio only one has to be determined and the other is automatically given by the ratio.

Using the bi-linear loss function and the decision rule does not lead to an analytical expression for the posterior expected loss as a function of the number of performed tests. This can be contributed to the fact that the expectation that need to be calculated cannot be performed analytically. Using a sampling techniques to calculate this expectation is implemented in a procedure in the next paragraph. As a new procedure needs to be written for this, it is not applied to the simplified timber case but straight to the (general) timber case.

Numerical solution

This paragraph will show how the ‘fixed size test plan’ can be determined in the timber case using sampling techniques and the analytical approach or the BBN approach. In order to perform the update analytically the \mathcal{NIG} -model is used. Also a new loss function is used which has an actual interpretation.

The following input has been received from the timber producer to define the decision problem. Table 7.2 gives the information needed to define the utility function. It is also given that the price difference between timber from two adjoining strength classes is 10 percent. The classification for the timber under investigation is given in Table 7.3. The prior information for this kind of timber is given as $m = 80, \Lambda = 8, \alpha = 8, \beta = 0.0005$.

Table 7.2: Input for decision problem

Description	Value
Batch size	$100m^3$
Beam size	$0.1m \times 0.2m \times 4m$
Prior price	$500 \frac{EUR}{m^3}$
Cost of manhours	$85EUR$
Test time	1 hour

Table 7.3: Classification of characteristic strength (NEN6760)

Class	I	II	III	IV	V	VI	VII
Strength	0-30	30-35	35-40	40-50	50-60	60-70	70-200

Using the give information makes it possible to define the decision problem in format that it can be solved by the fixed size test plan approach. Using the producers approach to formulate the decision problem leads to the following considerations. The batch of timber($100m^3$) is graded and then tests can be used to improve the grading. The timber is graded by calculating the characteristic strength R^{Char} from the (predictive) distribution as the 5-percentile point. This point is then graded into a strength class with Table 7.3. The graded characteristic strength of the batch (R^G) is given as the left bound of the class. Notice that when the actual characteristic strength (R^{Char}) is below 30 it is graded as strength class I. Timber in strength class I has a graded characteristic strength of $R^G = 0$ and is thus useless.

From a producers point of view a clear interpretation can be given to the loss function and the testing costs. The costs of destructively testing a beam from the batch is given as the price of the beam and the cost of the man-hours needed to perform the test. This implies the cost for this kind of test given as

$$C(n) = 125n. \quad (7.4.8)$$

The decision loss can be given as a function of the determined strength class of the batch. If the strength class is not changed after testing the loss equals 0. If the expected strength class after testing is one class lower the loss is given as the batch size times the difference in sale price of the timber in the different strength classes. The same holds for the expected gain when the expected strength class is one larger the loss is given as the batch size times the difference in sale prices¹¹. Note that the maximal loss is given to when the strength class becomes I, then it equals the batch size times the price of the prior strength class, this as the batch cannot be sold. Now the decision loss and the cost of testing have been determined the loss function has been determined which means that the expected loss for the fixed size test plan can be determined. This is done using the pseudo code given below. The result will be a plot of the expected loss against the number of performed test, from which the optimal test size can be determined.

Pseudo Code ‘Fixed Size Test Plan’ Using Sampling

Initialization

- Define constants,

¹¹Notice that this difference is now negative thus there is no loss, gain.

- Define prior information,
- Define strength classes,
- Define utility function,
- Process prior information,

Computation (loop over number of runs(i))

- Loop over number of samples(j)
 - Sample $\eta_{i,j}, \mu_{i,j}$,
 - Loop over number of tests(k)
 - * Sample destructive data, $d_{i,j,k}$
 - * Calculate sampled characteristic strength(perform update) $r_{i,j,k}^{Char}$,
 - * Classify sampled characteristic strength $r_{i,j,k}^G$,
- Calculate expected loss,

Post processing

- Plot expected loss against number of test for each run.

This pseudo code is implemented in MATLAB as, ‘SampleME8.m’(see Appendix C). The results of this m-file are displayed in Figure 7.5. From this figure it is determined that four tests should be performed. Multiple curves have been plotted in this figure to display the convergence of the sampling used to determine the expected loss.

The m-file that produced Figure 7.5 uses a special approach to sample destructive tests. The pseudo code also showed this, the parameters M, H are sampled and then used to generate samples for the destructive test results. It is important that it is done in this order as given these sampled parameters M, H the test results are independent. Then on the basis of these sampled test result and the prior information a sampled characteristic strength is determined. In ‘SampleME8.m’ an analytical approach is used for this, by application of the \mathcal{NIG} -model¹². By use of this model the characteristic strength can be given in terms of the parameters of the prior distribution and the sampled test result, this is formulated in the following proposition.

Proposition 7.4.1. *The characteristic strength of the timber (0.05 percentile point) when using the \mathcal{NIG} -model is given in the parameters of the \mathcal{NIG} -model, see Equation¹³ (5.1.7), as*

$$R^{Char} = m^* + \sqrt{\frac{1 + \Lambda^*}{\beta^* \alpha^*}} \Psi_T^{-1}(0.05, 2\alpha^*), \quad (7.4.9)$$

in which $\Psi_T^{-1}(0.05, 2\alpha)$, denotes the inverse cumulative distribution function of a standard student’s t -distributed variable.

Proof. The proof of this proposition is listed in the Appendix A.3, which constitutes of rephrasing Equation (5.1.1). □

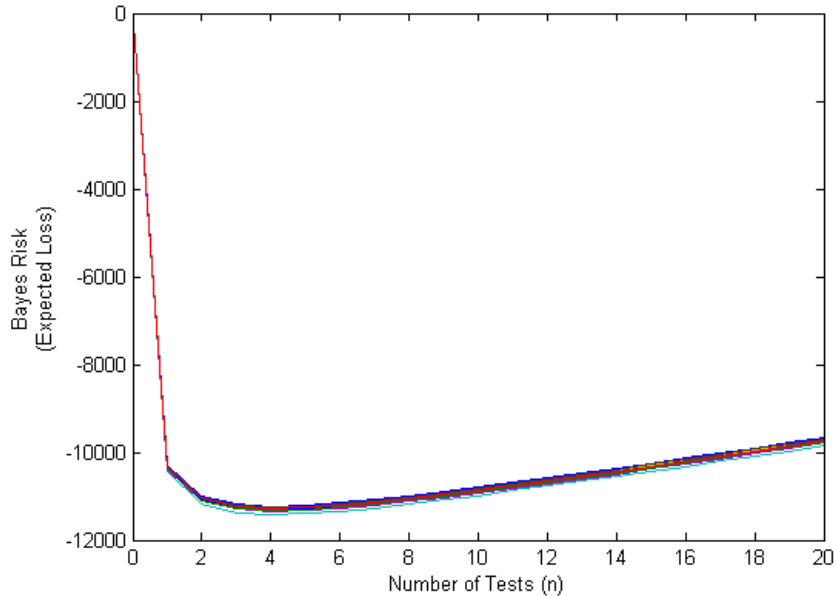


Figure 7.5: A plot of the expected loss against the number of tests

The use of an analytical approach to perform the update and thus determine the sampled characteristic value is not necessary. The updating can also be done otherwise, numerically or by a BBN. For the timber case using the \mathcal{NIG} -model also a MATLAB program was created that uses a BBN approach (HUGIN) to perform the update. This program is added in the Appendix C, as ‘SampleMEHugin4.m’. Note that this m-file is implemented using the same pseudo code only the step calculate sample characteristic strength is done by HUGIN. By comparisons of the results of both the codes using HUGIN and the analytical approach it is concluded that they give the same results. The main reason for adding this code to the appendix is that it can easily be applied in other situations and other decision problems. Using the BBN approach for the updating also has another advantage. The graphical model defined in the BBN approach essentially gives the sampling order, as only prime nodes are sampled. This is discussed in Chapter 8.

7.4.4 Application VOI

Using the value of information to create an optimal test plan needs a more mathematical formulation of the decision problem. The decision problem to determine the optimal fixed size test plan can thus formulated as

\mathcal{S} , the states of the world s_i denotes the state of the world, e.g. the prior characteristic strength;

\mathcal{F} , the set of acts f_n perform the n proposed tests;

¹²This model is defined in Chapter 4 and applied in Chapter 5.

¹³To determine the characteristic strength (R_{Char}) prior to testing use the parameters of the prior information.

f_0 do not perform the proposed test.

C the set of consequences Let the consequences be define over the set of $\mathcal{F} \times \mathcal{S}$, resulting in:

$c_{n,m}$ the decision has been made to perform n tests and the Bayesian update has been performed and using the test results leading to $c_{n,m}$, e.g. the (updated) predictive characteristic strength; c_0 the decision has been made not to perform the test.

The *Utility*-function has typically the following form:

$$U_{1,\dots,k}(c) = U(\mathbf{s}, f) - \sum_{i=1}^k \kappa_i \quad (7.4.10)$$

where \mathbf{s} is k -dimensional and κ_i denotes the costs of performing test i .

The following considerations derive the optimal test plan. A test is added to the test plan, i.e. is performed when

$$\nu(F|T_p \rightarrow T_p \cup T) \neq \nu(F|T_p) \quad (7.4.11)$$

Where T_p denotes the test plan. Which is equivalent to stating

$$\nu(F|T_p \cup T) > \nu(F|T_p) \quad (7.4.12)$$

This gives the following criteria for constructing a test plan:

Test 1 The first test is performed when

$$\nu(F|T_1) > \nu(F|\emptyset) = \nu(F) \quad (7.4.13)$$

Using the definition for the value of information in Equation (7.4.13) defines the following restriction on the cost of the first test.

$$\sum_{d_1 \in D_1} p(d_1) \max_{f \in F} \sum_{c \in C} p(f^{-1}(c)|d_1) U_1(c) > \max_{f \in F} \sum_{c \in C} p(f^{-1}(c)) U(c) \quad (7.4.14)$$

All the probabilities in Equation (7.4.14) are known/ given by the priors or and iff the test outcome is known. Thus by solving this equation for the cost of a test gives a condition on the costs of a test.

Test 2 The second test is performed when

$$\nu(F|T_1 = t_1, T_2) > \nu(F|T_1 = t_1) \quad (7.4.15)$$

By definition this means

$$\sum_{d_2 \in D_2} p(d_2|D_1 = d_1) \max_{f \in F} \sum_{c \in C} p(f^{-1}(c)|D_1 = d_1, D_2 = d_2) U_{1,2}(c) > \max_{f \in F} \sum_{c \in C} p(f^{-1}(c)|D_1 = d_1) U_1(c)$$

This equation needs to be solved for the outcome of test one giving a condition on the outcome of test one to either perform the second test or not.

This approach does not seem to give an feasible approach as for each utility and probability a predictive distribution needs to be calculated. This gives to much computations. Also the formulation of the decision problem in this setting does feel a bit unnatural. Possibly it can be formulated as accepting the sample or rejecting the sample as possible acts. This is then added as future research.

Chapter 8

General Method for Testing

8.1 The General Method

This chapter demonstrates the use of the theory derived in the previous chapters. It can be seen as an attempt to create a complete methodology which stipulates the use of testing/observations. The following general method is proposed

General Method for Testing

Problem definition

1. Define testing objectives and decision problem,
2. Create models,
3. Define primary variables,

Interpretation of test results

4. Apply Bayesian method,
5. Validation of Bayesian method and models,

Test plan

6. Define optimal test plan,

Evaluation

7. Interpret results.

There are two main steps in this method, respectively the fourth and sixth step. All other steps are preparations for these main steps or checks of these two main steps. The time when the first six steps are performed is different from the last step. The first six steps are performed before the tests are performed and in step seven the test results are available and are investigated. This is an essential difference. Note that for performing the seventh step the method defined in step four can be used. Also note that often this method will be performed in a loop. After the seventh step often a new decision problem is formed to determine if it is useful to perform additional test given the new

information gained from the tests. The result can be to perform even more tests, which does not imply that the first application of the method has failed. It only means that the information gained from the tests is more or at least different than was expected. The following section will show an application of the method.

8.2 Humidity Case

The case that will be used to demonstrate the proposed method for testing is referred to as the *humidity case*. It addresses the issue of determining the time when the humidity on the outside of a concrete wall is lower than a predefined level. This case has been derived from the following actual situation, a contractor wants to apply stucco to a wall. In order for the stucco to stay fixed to the concrete wall, the wall needs to be dry. Often after some time humidity from within the wall reaches the outside of the wall and affects the stucco. To avoid this stucco is applied after a pre-defined number of days. This in order to let the humidity from within the wall to dissipate. This sometimes creates unnecessary delay in the construction as the waiting period is far too long. Now testing is proposed to be used to determine when the wall is dry enough, and not use the pre-defined number of days. This in order to decrease the time that needs to be waited till the stucco can be applied.

In order to model the movement of the humidity a small simplification in the modelling of the humidity movement within the wall is used as it is modelled as one-dimensional. In Figure 8.1 a wall under investigation is displayed schematized displaying the mentioned simplification.

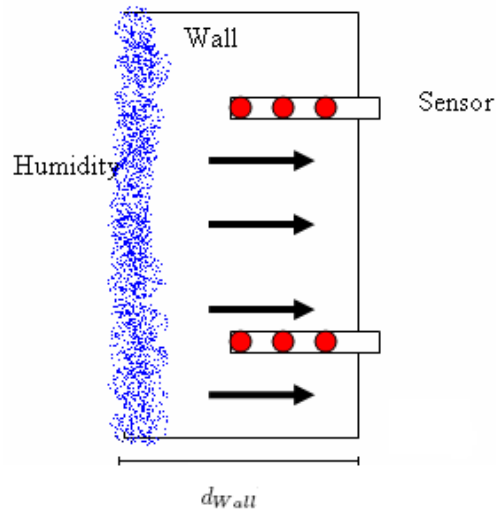


Figure 8.1: Schematized view of a wall under investigation

In this figure a distance (d_{wall}) is displayed, this is the average distance the humidity needs to

Table 8.1: Input for decision problem

Description	Value
Wall thickness	35mm
Fixed cost of performing tests	100EUR
Cost of performing a single measurement	15EUR
Sensor size	3 measurement points
Savings	500EUR/week

travel to reach the stucco¹. Also a sensor is displayed in Figure 8.1. This special sensor measures the humidity at different depths, referred to as measurement points (a(n) (observation) model for this sensor will be defined later). The use of this sensor defines the decision problem, which can be phrased as, how many measurements with this kind of sensors should be performed to optimal determine the time till the stucco can be applied. Now that the case is briefly introduced the general method for testing can be applied.

Define testing objectives and decision problem

In this case the objective is to investigate if it is possible to determine, with testing, the time when the humidity in the wall has dropped to an acceptable level. This analysis will take into account the expected benefits and costs of testing. The decision problem can be defined as: how many tests are 'ideally' needed to determine the time when stucco can be applied. Notice that the time when a test is performed is taken as fixed. Table 8.1 gives the parameters that specify the construction under investigation and the decision problem.

Create models

To model the movement of the humidity in the concrete wall the following relation is used to describe the distance the humidity travels in time.

$$D(t) = R_v \sqrt{t}. \quad (8.2.1)$$

In which time is denoted as t in [days] and the humidity intrusion rate as R_v in [$mm/days^{\frac{1}{2}}$]. Note that R_v is assumed to be time independent. From experience gained from previous experiments R_v is assumed a-priori to follow a log-normal distribution² with mean $2[mm/day^{\frac{1}{2}}]$ and a standard deviation of $1[mm/day^{\frac{1}{2}}]$. This relation is used to describe the distribution over the time needed for the wall to be dry at the outside of the wall. The humidity that is present in the concrete wall is modelled to travel (on average) a distance d_{Wall} . Together with Equation (8.2.1) this gives the following model for time needed by the wall to be dry on the outside of the wall,

$$T_{Dry} = \left(\frac{d_{Wall}}{R_v} \right)^2. \quad (8.2.2)$$

This in terms of the Bayesian method is the 'forward model', see Chapter 3.

¹In case of solid concrete wall this distance can be thought of as half the wall thickness.

²The parameters of this distribution are $M = 0.5816$ and $V = 0.4724$.

As mentioned earlier the observations/measurements are done with a sensor, this sensor measures the specific resistance in the concrete at three different depths (positions) in the wall. The depths where the measurements are taken are placed at a fixed distance from each other. Thus a single observation is represented as a vector containing specific resistance measured at each depth. If this sensor has q measurement points this is denoted as

$$\underline{\Omega}_m = [\Omega_m^1 \Omega_m^2 \dots \Omega_m^q]' \quad (8.2.3)$$

The following three (observation) models describe the relation³ between the measurements and the humidity intrusion rate (R_v).

Measurement accuracy

$$\Omega_m^i = \Omega_r^i \varepsilon_i. \quad (8.2.4)$$

Conversion error

$$\Omega_r^i = \frac{1}{B} \log \left(\frac{V^i}{A} \right). \quad (8.2.5)$$

Humidity relation

$$V^i = v_0 - \frac{(v_0 - v_c)}{R_v \sqrt{t}} x_i. \quad (8.2.6)$$

The first two models describe the error in the measurements, the last the relation between humidity at a location (x) and R_v . The *measurement accuracy* describes the relation from the measured specific resistance to the actual specific resistance at that depth in the construction, i.e., the imperfection in the measurement. Where respectively $\varepsilon_i \sim \mathcal{N}(1, 0.05)$ denotes the (multiplicative) error, Ω_r^i the real specific resistance $[\Omega m]$ at depth i , and Ω_m^i the measured specific resistances $[\Omega m]$ at depth i . The *conversion error* models the error induced by the (empirical) relation between specific resistance and humidity. As a result this error is the same for all measurement points. A and B both have a-priori normal distributions⁴ with respectively mean $150.18 [kg/m^3]$, $1.21 * 10^{-5} [\Omega^{-1} m^{-1}]$ and variance 0.05 and 0.1. Respectively the *humidity relation* described by Equation (8.2.6). This equation describes the relation⁵ between the humidity (V) present in the wall and the humidity intrusion rate (R_v).

When multiple measurements are performed they are thus performed in ensembles of three measurements. The models given above do not (yet) take into account (spacial) correlation. Therefore multiple measurement can only be used to learn about the conversion error and thus better determine the humidity intrusion rate (R_v). This can be defended by assuming homogenous humidity over the surface of the wall. Also a model for the cost of testing is needed to solve the decision problem, which in this case is a linear cost function,

$$C(n) = C_S + nC_T. \quad (8.2.7)$$

In which C_S denotes the (fixed) setup cost for testing describing the costs made to develop or buy the sensor, C_T the cost of performing a test and analyzing the data received from a sensor. The decision that needs to be made is to define a number of tests to perform at a specific point of time.

³Note that this defines the likelihood function.

⁴As a first approximation the errors are modelled as independent.

⁵Using this relation to describe the relation between the humidity present in the wall and the rate of intrusion is not completely correct for a wall under inspection. This model was derived and implemented in this case during the process of defining the Bayesian method and was then fully applicable. However the case specifics changed when define the decision problem and this model was still used as no time was left to create another more appropriate model. This does not have important consequences for the demonstration of the method. Only the results of the case are not completely true as the update is not done correct.

Table 8.2: Conditional probability table of node error1 in HUGIN

Expression	Normal(1,0.05 ²)
-inf - 0.95	0.158655
0.95 - 1	0.341345
1 - 1.05	0.341345
1.05 - 1.1	0.135905
1.1 - inf	0.02275

Define Primary Variables

Now all models have been defined, the stochastic variable to which the Bayesian method will be applied needs to be defined. From all the models, R_v is identified as the random variable to which the Bayesian method will be applied. Still the random variable of interest is T_{Dry} . Actually during the derivation of the models these variables were already identified as the primary variables. How else would all the models be created? This step is placed in the method to emphasize that this need be done. In practice this step is integrated into the previous step.

Apply Bayesian Method

We will start the Bayesian Method by using the graphical representation of the method and substituting into it the actual models and stochastics. This will structure the process. As first the relations between the stochastic variables are defined. We start of with the general graphical formulation given in Figure 3.1, now we define the proper stochastic variables which results in Figure 8.2(a) (displayed on page 66). Then we define the predictive model in the graphical representation. The models that are inserted are shown in blue boxes. Next the observation models are inserted for one measurement point, again blue boxes are used. Finally an entire sensor is modelled in Figure 8.2(d).

This graphical representation is used to model the Bayesian method with a BBN. This is of course not a coincidence, the BBN-approach is chosen as it can easily be constructed from this graphical model. The graphical part of the BBN is almost the same as the graphical model displayed by Figure 8.2(d). The main difference is that names of the nodes are displayed and not the complete models. These models are used in the quantitative part of the BBN, defining the arcs in the BBN. The result of this graphical part of the BBN in HUGIN is displayed in Figure 8.3 (on page 67).

Defining the graphical part of this BBN is only the start of the modelling using a BBN, now the quantitative modelling needs to take place. As set out by the Bayesian method first the prior information needs to be entered. This means that the prior distribution for all primary nodes, i.e. $R_v, A, B, Error_1, Error_2, Error_3$ needs to be defined. Notice that this is actually implementing the information given in step one of the method. For example it was formulated that the measurement accuracy can be described with Equation (8.2.4) in which the error-term nodes ($Error_1, Error_2, Error_3$) follow a normal prior distribution. This is then implemented in the quantitative modelling part of the BBN. As HUGIN performs discrete calculation a discretisation needs to be defined for this node (and for all other nodes in the BBN). As the prior distribution has mean 1 and std.dev. 0.05 the discretisation displayed in Table 8.2 is used. Now the prior distribution can be entered⁶, in this case a model is used⁷ in HUGIN, resulting in the following CPT, see Table 8.2.

As can be seen in Table 8.2 two states have been added to create a full support for the distribution of the error-term. This is only to ensure a correct definition of the problem in HUGIN.

⁶This is also done with MATLAB see the Appendix C all m-files containing Hum.

⁷This is not necessary, tables can also be filled manually.

The second step in the Bayesian method is to define the likelihood. This is done in HUGIN by defining models for the CPT of the nodes that appear in the observation models. When this is done the entire BBN is defined and can be used to perform the update and perform inference, which concludes the Bayesian method. The entire network was created in MATLAB and the code can be found in the Appendix C as ‘HumCreateNW.m’.

Validation of Bayesian Method and Models

This next step in the method is to ensure that the previous steps have been performed correctly. By entering possible observations and subsequently performing the update the output of the Bayesian method is checked for errors. Often small adjustments in previous steps in the method need be made, e.g. discretisations need to be adjusted. Also it serves as a check to see whether the important factors have been identified and it shows what kind of testing should be done.

In this application of the general method we have proposed to use HUGIN (and MATLAB) to perform the Bayesian method, as this is a robust approach and easy to implement as it is constructed in a very natural way. It can sometimes be more efficient to use other methods discussed in chapter 4. In the humidity case also the direct numerical integration approach has been used. This to check whether HUGIN was used correctly to perform the update and to see the difference between the two applications. This is omitted in the appendix in subsection A.5.

Define Optimal Test Plan

In this step of the method the test plan has to be derived. This means solving the decision problem defined in the first step. In order to do this three possible approaches were given in Chapter 7. During the thesis research a combination of the ‘Decision tree’ for visualization purposes and ‘Fixed size test plan’ approach for calculations was found to be very useful, the latter of these is thus used. The following parameters were gathered for the decision problem in the first step. These parameters are used to define the utility(loss) function. This function will be used to measure the effects(gain) of performing tests and the costs. The costs are modelled as being independent of the test result and only dependent on the number of performed tests. The fixed cost of performing test are given as $100EUR$. The (variable) cost of performing a single test are given as $3 * 15EUR = 45EUR$ a test. These variable costs are the cost of performing a measurement, e.g. cost of transmitting test results from sensor to computer or power needed to use the sensor, and costs for interpreting the test results. These numbers can be substituted into Equation (8.2.7) leading to the cost function.

In order to measure the effect(gain) of performing a test the following considerations are used. Note that the following reasoning can be seen as using the ‘producers approach’, see sub-sections 7.2 and 7.4. The forward model is used to determine the time till the wall is dry, see Equation (8.2.2). Using this model gives a distribution over the time till the wall is dry. The regulations do not give a distribution over the time till the wall is dry but give a number of days to wait till the wall is dry (T_{Dry}^{rule}). Using this pre-defined number of days that are needed to wait, which in this case is 90 days after construction of the wall is complete, gives a bounded region for T_{Dry} . The contractor that needs to apply the stucco does his scheduling of his work at the beginning of the week. This implies that he is indifferent if T_{Dry} changes within the range of a week as he cannot change his schedule to act upon this information. This then defines the discretisation for the domain of T_{Dry} as 0 to 84 with steps of 7 days. Which means in HUGIN the following discretisation given in Table 8.3. If the time the contractor has to wait shortens with a week he makes $500EUR$.

As mentioned, the result of using a test is reflected in the distribution of T_{Dry} via the predictive

Table 8.3: Discretisation for T_{Dry} in HUGIN

-inf-0	0-7	...	77-84	84-inf
--------	-----	-----	-------	--------

(forward) model. The regulations do not specify a distribution but give a fixed number, T_{Dry}^{rule} . In order to compare the effects of the test result the predictive distribution over T_{Dry} is used to give a new point estimate of the time till the wall is dry. This is referred to as using a decision rule. This decision rule is thus needed to measure the effects of testing, i.e. to formulate a loss function. In this case the 80-percentile point⁸ of the predictive distribution for T_{Dry} is used as $T_{Dry}^{pred}(n)$. This estimate gives the days one has to wait before applying the stucco according to the prior information and the n test results. Using this information gives the following loss(utility) function.

$$L(n) = 100 + n35 - 500 \left(T_{Dry}^{rule} - T_{Dry}^{pred}(n) \right) 1_{T_{Dry}^{pred}(n) < T_{Dry}^{rule}} \quad (8.2.8)$$

In which $1_{A < B}$ is the indicator function which means that it is 1 if $A < B$ and 0 if $A > B$. Equation (8.2.8) clearly shows that reducing the waiting time with a week results in 500EUR at the cost of performing the needed tests, and also that if the tests lead to an estimate that is equal or larger then what the regulations prescribe no gain or loss is achieved as the contractor will use the pre-described⁹ T_{Dry}^{rule} .

Figure 8.4 has been created using the program ‘HumFinal.m’ which is added in the appendix. This figure is used to determine the optimal number of tests. Figure 8.4 clearly shows a minimum when performing nine tests. Again multiple curves have been plotted to show that the sample size used is sufficient large.

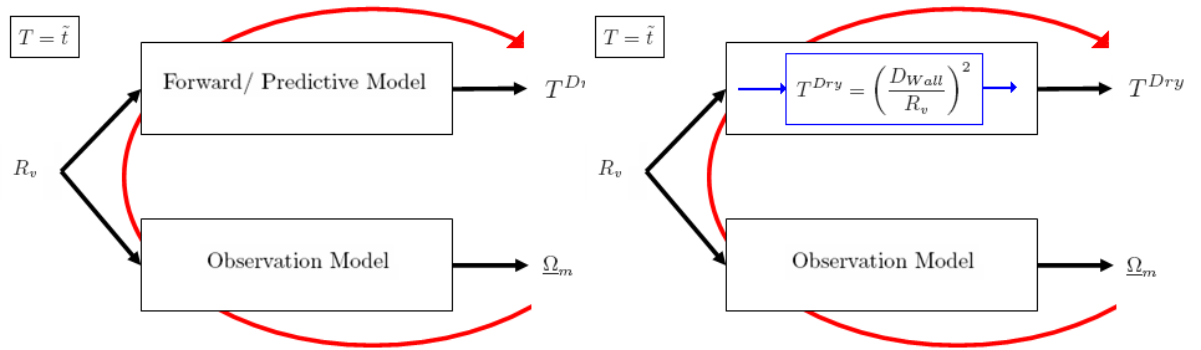
Interpret Results

As the test plan has now been derived the tests can be performed. For interpreting the test results (observations) the Bayesian method can be re-used, which was already implemented prior to testing thus the procedures defined then can be re-used for this. Depending on the results the method is finished or initiated again taking into account the information gained from the testing.

If an adaptive test plan has been created there will be a loop from this step to previous steps until the testing is completely done as defined by the adaptive test plan (stopping criteria reached). Thus completing the method.

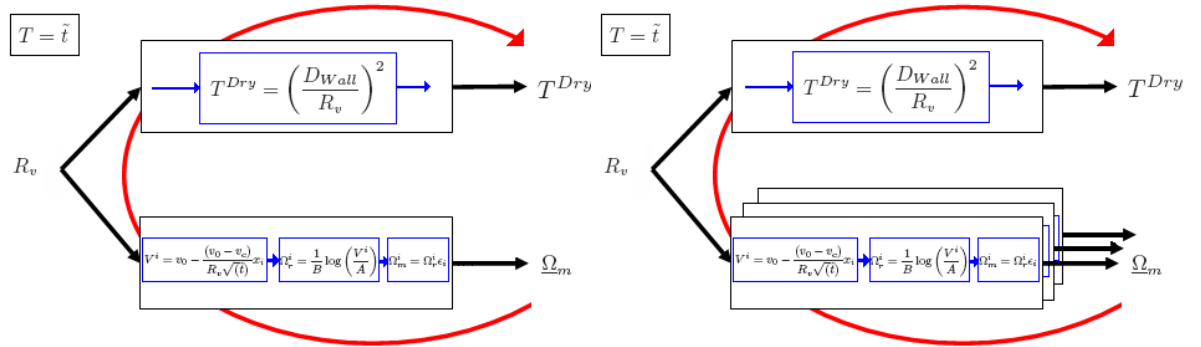
⁸In this case the 80-percentile point was chosen to create a decision rule. Another approach can be to use the prior information on R_v and calculate the prior predictive (cumulative) distribution function of T_{Dry} . Then check which point corresponds with the one given by the regulations and use that as the decision rule. This is however subject to the prior information used. In cases in which no clear decision rule can be found and strong prior information is available it can give a solution.

⁹Which makes him non liable if the stucco fails as he has used the rules, thus no extra loss. Although the contractor did know that the stucco would probably be effected by the humidity indicated by the (ignored) test results.



(a) Stochastics defined

(b) Forward/predictive model defined



(c) Observation model defined

(d) Multiple Measurements

Figure 8.2: Building the graphical representation of the Bayesian method

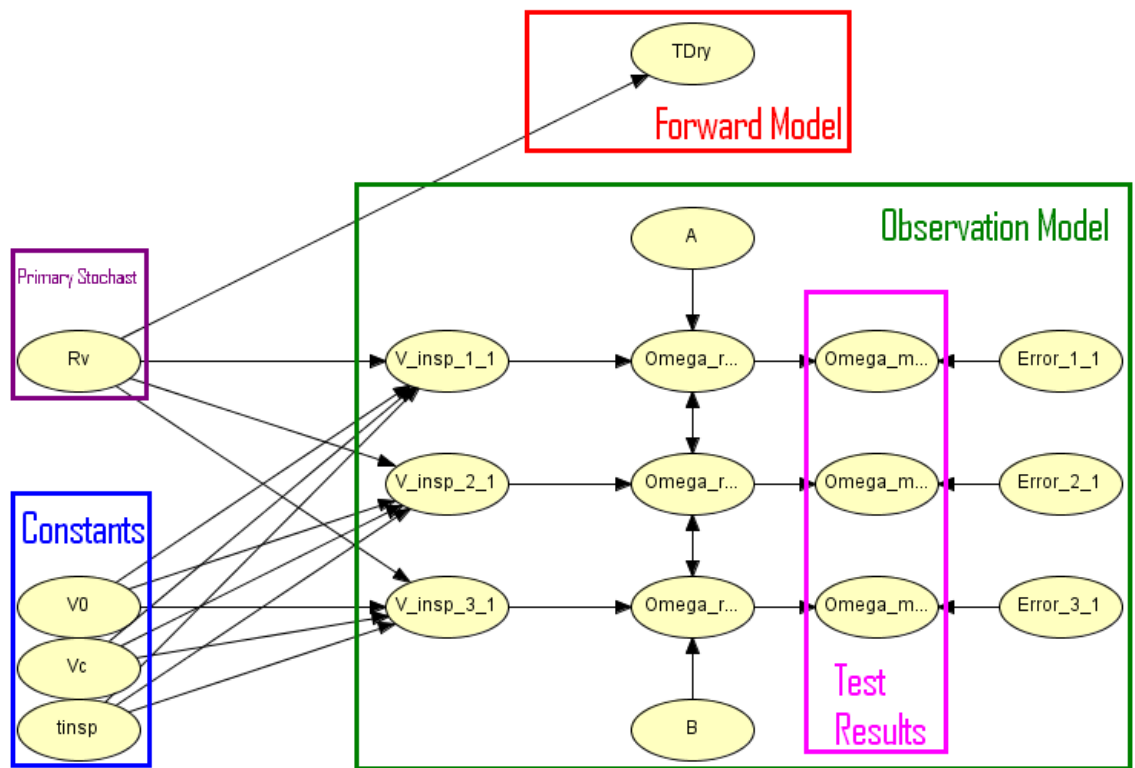


Figure 8.3: The graphical model for the humidity case

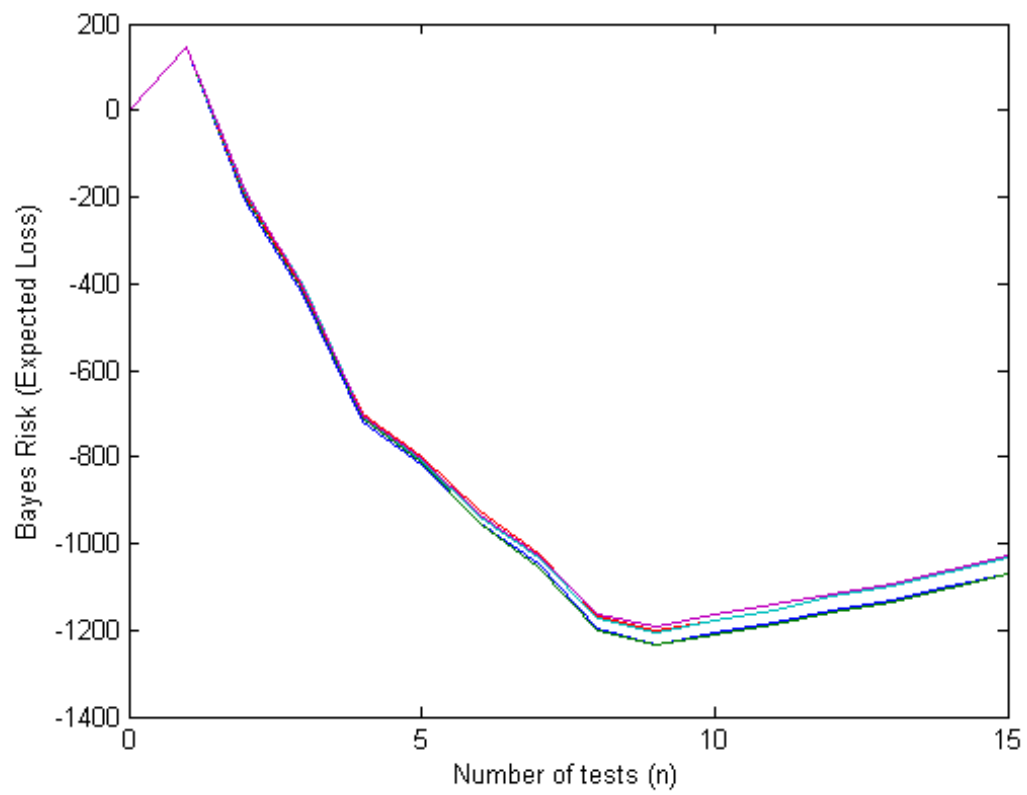


Figure 8.4: Results of the fixed size test plan approach in the humidity case

Chapter 9

Conclusions, Recommendations and Further Research

9.1 Conclusions

In this thesis a general method was defined that tells how to test. The method was shown to solve the problem of defining the number of tests that need to be performed. This was done by taking into account the expected information received from a test result and the cost of performing a test.

The general method for testing consists of two main elements namely the interpretation of test results and the decision what test results to obtain. The Bayesian method was applied for the interpretation of the test results and Savage's decision Theory was used for the decision part. Four different approaches to apply the Bayesian method were discussed in this thesis. It can be concluded that the Bayesian belief network (BBN) approach is the best approach to use in a general setting. This as the BBN approach contains a graphical model which simplifies communication of the used models. Another advantage of using the BBN approach is that it requires minimal mathematical knowledge of the user, multiple software packages(i.e. HUGIN) can be used for that. These software packages can also be helpful developing a numerical tool for the general method as they can be used. Also in the BBN-approach after the problem is defined it can easy and fast process different (new) data within the model.

The decision part in the general method consists of two aspects, namely definition and calculation. For the calculation of the solution to the decision problem, which means deriving an optimal number of tests to perform, three approaches have been proposed, namely the decision tree, fixed size test plan and value of information approach. It can be concluded that the fixed size test plan easily calculates the solution and that the decision tree can best be used for visualizing the performed calculations.

Two cases, the timber and the humidity case have been discussed and solved with this general method. The result of this is a series of m-files that implement the method. These m-files were set up in a general formulation that they can easily be reused or adapted to other cases. Making the combination of all these m-files a blue print for a (numerical) tool to solve these decision problems. These m-files running in MATLAB also use the HUGIN-engine to perform the calculations. This was done by controlling HUGIN via MATLAB trough the HUGIN API, a report was added to the appendix explaining how this is done.

9.2 Recommendations

The following recommendations are made on the use of the Bayesian method. It was concluded that the best approach to perform the Bayesian method was the BBN approach. This is only true as it is easy understood and very robust. In some cases using one of the other four given approaches (see Chapter 4) can very well lead to faster and more detailed results.

Another recommendation is that when defining models also bear in mind that the \mathcal{IG} -model can be used and not only the \mathcal{N} -model. As was shown when discussing the analytical approach a (computational fast) conjugate analysis is sometimes possible when using the \mathcal{NIG} -model where it was not possible with the \mathcal{N} -model.

9.3 Further Research

This section sets forth the research that should be done to improve or further develop the general method of testing. The following research topics are given for the general method. The method should be applied to more cases to see whether more steps need to be added to the method. The method can also be expanded to handle multiple decision objectives. In this thesis only the number of tests were investigated as a decision objective but other objectives might be included in the method, like what location should be used for testing. This will need further refinement of the models as spatial correlation between test etc. Also further work can be done to create a numerical tool implementing the general method. This can be done by further developing the m-files that have been added to this report into a full numerical tool.

Also some further research can be done on the the Bayesian method. The BBN approach can be further refined in many ways. For instance the BBN approach can be refined to also include continuous BBN's. This will in some cases make the modelling much more natural. Also the BBN approach as discussed (using discrete BBNs) can be further automated, by automating the discretisation definition. In this thesis all nodes have been discretised by visual inspection of the results over the discretisation, ideally this will be done by a program and the user need not to worry about this.

The Bayes linear approach, that did not seem to work in a general setting, should be further investigated. This in order to fully understand the reasons it did not perform well in a general setting. If this can be found the approach can be improved to make it suitable for a general application or by given bounds on the estimates it gives.

Further research needs to be done on defining the decision problem in a general case. This as finding a utility function in case of a real problem has shown to be very difficult, even though two approaches for this have been given, respectively designers and producers approach. As well defining the decision problem as a sequential decision problem needs to be investigated. This can possibly give a more accurate solution to the decision problem posed or speed up calculations.

The approaches that were used to calculate the solution to the decision problems can also be refined and evolved. In thesis only one-dimensional decision problems were solved, like how many test should be performed. The given approaches can easily be used in higher dimensional decision problems, like which combination of test type A en B should be used. In order to do this the models should be upgraded to handle the dependency between different test types and the now visually choosing the optimum needs to be replace by calculations¹.

The value of information approach needs to be further researched. During the thesis research this approach did not seem to work and in the authors opinion it was due to the problem definition.

¹Visually determining the optimum is not possible for decision problems involving higher dimensions then two.

Appendix A

Appendices to the Text

A.1 Proof Multiple Destructive Observations

Proof Proposition 5.1.2. In order to prove that the predictive distribution of R is student-t distributed, when n destructive measurements are used, proposition 5.1.1 is used. Therefore it is sufficient to proof that the posterior distribution over M, H is $\mathcal{NTG}(\alpha^*, \beta^*, m, \Lambda)$ when using n destructive measurements. This proof goes along the lines of proof 4.1.1 on page 16. The likelihood function of n -destructive measurements in the timber case is give as

$$\begin{aligned}
 L(\underline{\delta}) &= f_{\underline{D}|M,H}(\delta_1 \dots \delta_n; \mu, \eta) \\
 &= \prod_{i=1}^n f_{D_i|M,H}(\delta_i, \mu, \eta) \\
 &= (2\pi\eta)^{-\frac{n}{2}} e^{-\frac{1}{2\eta} \sum_{i=1}^n (\delta_i - \mu)^2}.
 \end{aligned} \tag{A.1.1}$$

Substituting Equation (A.1.1) and the joint-distribution function in Bayes Rule (1.2.2) leads to.

$$\begin{aligned}
 f_{M,H|\underline{D}}(\mu, \eta; \underline{\delta}) &= C_n (2\pi\eta)^{-\frac{n}{2}} e^{-\frac{1}{2\eta} \sum_{i=1}^n (\delta_i - \mu)^2} \frac{\eta^{-\alpha^* - \frac{3}{2}} \beta^{* - \alpha^*}}{\sqrt{2\pi} \Gamma(\alpha^*) \Lambda^{*\frac{1}{2}}} e^{\left[-\frac{1}{2} \frac{(\mu - m^*)^2}{\eta \Lambda^*} - \frac{1}{\eta \beta^*}\right]} \\
 &= \tilde{C}_n \eta^{-\alpha - \frac{3}{2} - \frac{n}{2}} e^{-\frac{1}{2\eta\Lambda} \left[(\mu - m)^2 + \Lambda \sum_{i=1}^n (\delta_i - \mu)^2 \right]} - \frac{1}{\eta\beta}
 \end{aligned}$$

where C_n, \tilde{C}_n are appropriate normalizing constants. Define $\alpha^* = \alpha + \frac{n}{2}$, substitution of α^* and re-arranging terms results in¹

$$f_{M,H|\underline{D}}(\mu, \eta; \underline{\delta}) = \tilde{C}_n \eta^{-\alpha^* - \frac{3}{2}} e^{-\frac{1+n\Lambda}{2\eta\Lambda} \left[\left(\mu - \frac{m+\Lambda \sum \delta_i}{1+n\Lambda} \right)^2 - \frac{(m+\Lambda \sum \delta_i)^2}{(1+n\Lambda)^2} + \frac{m^2 + \Lambda \sum \delta_i^2}{1+n\Lambda} \right]} - \frac{1}{\eta\beta}. \tag{A.1.2}$$

¹Notice that the index of the summations have been dropped. This has been done for typographical reasons it should show $i = 1$ up to n .

Also define the following variables $\Lambda^* = \frac{\Lambda}{1+n\Lambda}$, $m^* = \frac{m+\Lambda\sum\delta_i}{1+n\Lambda}$, and

$$\beta^* = \frac{2\beta(1+n\Lambda)}{-\beta(\Lambda(\sum\delta_i)^2 + 2\sum\delta_im - nm^2 - (1+\Lambda n)\sum\delta_i^2) + 2(1+n\Lambda)}.$$

which results in

$$f_{M,H|\underline{D}}(\mu, \eta; \underline{d}) = \frac{\eta^{-\alpha^* - \frac{3}{2}} \beta^{*-\alpha^*}}{\sqrt{2\pi}\Gamma(\alpha^*)\Lambda^{*\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m^*)^2}{\eta\Lambda^*} - \frac{1}{\eta\beta^*}}. \quad (\text{A.1.3})$$

Equation (A.1.3) clearly shows that the posterior distribution for M, H is $\mathcal{NIG}(\alpha^*, \beta^*, m^*, \Lambda^*)$ which concludes the proof. \square

A.2 Calculations and Proofs to Propositions Concerning Comparison Bayes Linear

Proof Proposition 5.4.1. The definition of the expectation gives us the following integral.

$$\mathbb{E}(R) = \int r f_R(r) dr \quad (\text{A.2.1})$$

Application of LTP results in.

$$\begin{aligned} &= \iiint r f_{R|M,H}(r; \mu, \eta) f_{M,H}(\mu, \eta) d(r, \mu, \eta) \\ &= \iiint r \frac{1}{\sqrt{2\pi}\eta} e^{-\frac{1}{2\eta}(r-\mu)^2} \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(r, \mu, \eta) \\ &= \iint \mu \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(\mu, \eta) \\ &= \int m \frac{\eta^{-\alpha-1} \beta^{-\alpha}}{\Gamma(\alpha)} e^{-\frac{1}{\eta\beta}} d\eta \end{aligned}$$

Now the following change of variables is performed to transform this integral to a gamma function; $\frac{1}{\eta\beta} = \xi$.

$$\mathbb{E}(R) = \frac{m}{\Gamma(\alpha)} \int \xi^{\alpha-1} e^{-\frac{1}{\xi}} d\xi = m \quad (\text{A.2.2})$$

□

Proof Proposition 5.4.2. Substitute D, δ for R, r respectively in the proof of proposition (5.4.1). □

Proof Proposition 5.4.3. By definition the following expression is available for the variance: $\text{Var}(R) = \mathbb{E}(R^2) - (\mathbb{E}(R))^2$. The right hand side of this expression has two parts which will be calculated sequentially below.

$$\begin{aligned} \mathbb{E}(R^2) &= \int r^2 f_R(r) dr \\ &= \iint r^2 f_{R|M,H}(r; \mu, \eta) f_{M,H}(\mu, \eta) d(\mu, \eta, r) \\ &= \iiint r^2 \frac{1}{\sqrt{2\pi\eta}} e^{-\frac{1}{2\eta}(r-\mu)^2} dr \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(\mu, \eta) \\ &= \iint (\eta + \mu^2) \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(\mu, \eta) \\ &= \iint \frac{\eta^{-\alpha-\frac{1}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(\mu, \eta) + \iint \mu^2 \frac{\eta^{-\alpha-\frac{3}{2}} \beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda} - \frac{1}{\eta\beta}} d(\mu, \eta) \\ &= \int \frac{\eta^{-\alpha} \beta^{-\alpha}}{\Gamma(\alpha)} e^{-\frac{1}{\eta\beta}} d\eta + \int (\eta\Lambda + m^2) \frac{\eta^{-\alpha-1} \beta^{-\alpha}}{\Gamma(\alpha)} e^{-\frac{1}{\eta\beta}} d\eta \\ &= (1 + \Lambda) \frac{\beta^{-\alpha}}{\Gamma(\alpha)} \int \eta^{-\alpha} e^{-\frac{1}{\eta\beta}} d\eta + \frac{m^2 \beta^{-\alpha}}{\Gamma(\alpha)} \int \eta^{-\alpha-1} e^{-\frac{1}{\eta\beta}} d\eta \end{aligned}$$

Again the following transformation is used $\frac{1}{\eta\beta} = \xi$.

$$\begin{aligned} \mathbb{E}(R^2) &= \frac{1 + \Lambda}{\beta\Gamma(\alpha)} \int \xi^{\alpha-2} e^{-\xi} d\xi + \frac{m^2}{\Gamma(\alpha)} \int \xi^{\alpha-1} e^{-\xi} d\xi \\ &= \frac{1 + \Lambda}{\beta(\alpha - 1)} + m^2 \end{aligned} \quad (\text{A.2.3})$$

The second term on the right hand side is given by proposition (5.4.1). Substituting these two results concludes the proof. □

Proof Proposition 5.4.4. Substitute D, δ for R, r respectively in the proof of proposition (5.4.3). □

Proof Proposition 5.4.5. In order to derive $\text{Cov}(R, D)$ the following integral needs to be calculated.

$$\mathbb{E}(RD) = \iint r\delta f_{R,D}(r, \delta) d(r, \delta) \quad (\text{A.2.4})$$

This can be done via application on LTP and using the priors, e.g.

$$\begin{aligned}
\mathbb{E}(RD) &= \iint r\delta \iint f_{R,D|M,H}(r, \delta; \mu, \eta) f_{M,H}(\mu, \eta) d(\mu, \eta) d(r, \delta) \\
&= \iint r\delta \iint f_{R|M,H}(r; \mu, \eta) f_{D|M,H}(\delta; \mu, \eta) f_{M,H}(\mu, \eta) d(\mu, \eta) d(r, \delta) \\
&= \iiint r\delta \frac{1}{2\pi\eta} e^{-\frac{1}{2\eta}((r-\mu)^2+(\delta-\mu)^2)} \frac{\eta^{-\alpha-\frac{3}{2}}\beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda}-\frac{1}{\eta\beta}} d(\mu, \eta, r, \delta)
\end{aligned}$$

The order of integration will now be changed. First integration over r than over δ will be performed.

$$\begin{aligned}
\mathbb{E}(RD) &= \iiint \delta \frac{\eta^{-\alpha-\frac{5}{2}}\beta^{-\alpha}}{(2\pi)^{\frac{3}{2}}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2\eta}(\delta-\mu)^2} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda}-\frac{1}{\eta\beta}} \int r e^{-\frac{1}{2\eta}(r-\mu)^2} dr d(\mu, \eta, \delta) \\
&= \iiint \delta \mu \frac{\eta^{-\alpha-2}\beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2\eta}(\delta-\mu)^2} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda}-\frac{1}{\eta\beta}} d(\mu, \eta, \delta) \\
&= \iint \mu \frac{\eta^{-\alpha-2}\beta^{-\alpha}}{2\pi\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda}-\frac{1}{\eta\beta}} \int \delta e^{-\frac{1}{2\eta}(\delta-\mu)^2} d\delta d(\mu, \eta) \\
&= \iint \mu^2 \frac{\eta^{-\alpha-\frac{3}{2}}\beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu-m)^2}{\eta\Lambda}-\frac{1}{\eta\beta}} d(\mu, \eta)
\end{aligned}$$

Now first perform the integration over μ .

$$\begin{aligned}
\mathbb{E}(RD) &= \int \frac{\eta^{-\alpha-\frac{3}{2}}\beta^{-\alpha}}{\sqrt{2\pi}\Gamma(\alpha)\Lambda^{\frac{1}{2}}} e^{-\frac{1}{\eta\beta}} \int \mu^2 e^{-\frac{1}{2\eta\Lambda}(\mu-m)^2} d\mu d\eta \\
&= \int \frac{\eta^{-\alpha-1}\beta^{-\alpha}}{\Gamma(\alpha)} e^{-\frac{1}{\eta\beta}} (m^2 + \eta\Lambda) d\eta
\end{aligned}$$

Here a gamma-function can be recognized. Define $\xi = \frac{1}{\eta\beta}$, this results in

$$\begin{aligned}
\mathbb{E}(RD) &= \frac{m^2}{\Gamma(\alpha)} \int \xi^{\alpha-1} e^{-\xi} d\xi + \frac{\Lambda}{\beta\Gamma(\alpha)} \int \xi^{\alpha-2} e^{-\xi} \\
&= m^2 + \frac{\Lambda}{\beta(\alpha-1)}
\end{aligned}$$

Together with proposition (5.4.1),(5.4.2) this results in $\text{Cov}(R, D) = \frac{\Lambda}{\beta(\alpha-1)}$. \square

A.3 Proof Characteristic Value of the Timber

Proof Proposition 7.4.1. From Proposition 5.1.2 it follows that the predictive distribution of R is (generalized) student-t distributed. The (inverse) cumulative distribution function of is found by

transforming the probability density function into the standard form. Equation (5.1.1) shows the following pdf.

$$f_{R|\underline{D}}(r; \underline{d}) = \frac{\beta^{*- \alpha^*} \Gamma(\alpha^* + \frac{1}{2})}{\sqrt{2\pi(1 + \Lambda^*)} \Gamma(\alpha^*)} (-C_2)^{-\alpha^* - \frac{1}{2}}, \quad (\text{A.3.1})$$

with

$$C_2 = -\frac{1}{\beta^*} - \frac{(r - m^*)^2}{2(1 + \Lambda^*)}, \quad (\text{A.3.2})$$

in which the parameters $\alpha^*, \beta^*, m^*, \Lambda^*$ are that of the posterior distribution over² $M, H(\mu, \eta)$. Notice that C_2 can be restated as

$$\begin{aligned} C_2 &= -\frac{1}{\beta^*} \left(1 + \frac{\beta^* (r - m^*)^2}{2(1 + \Lambda^*)} \right) \\ &= -\frac{1}{\beta^*} \left(1 + \frac{\alpha^* \beta^*}{(1 + \Lambda^*)} \frac{(r - m^*)^2}{2\alpha^*} \right) \\ &= -\frac{1}{\beta^*} C_3. \end{aligned} \quad (\text{A.3.3})$$

Define $C_4 = \frac{\alpha^* \beta^*}{(1 + \Lambda^*)}$, substitution of C_3 and C_4 in Equation (A.3.1) results in

$$\begin{aligned} f_{R|\underline{D}}(r; \underline{d}) &= \frac{\beta^{*- \alpha^*} \beta^{*\alpha^* + \frac{1}{2}} \Gamma(\alpha^* + \frac{1}{2})}{\sqrt{2\pi(1 + \Lambda^*)} \Gamma(\alpha^*)} (C_3)^{-\alpha^* - \frac{1}{2}} \\ &= \frac{\sqrt{C_4} \Gamma(\alpha^* + \frac{1}{2})}{\sqrt{2\alpha^* \pi} \Gamma(\alpha^*)} \left(1 + C_4 \frac{(r - m^*)^2}{2\alpha^*} \right)^{-\alpha^* - \frac{1}{2}} \end{aligned} \quad (\text{A.3.4})$$

From which a standard student-t distribution can be recognized, with $\nu = 2\alpha^*$ degrees of freedom, adjusted by C_4 and shifted over m^* , i.e.,

$$f_{R|\underline{D}}(r; \underline{d}) = \sqrt{C_4} \psi_T(C_4(r - m^*), 2\alpha^*), \quad (\text{A.3.5})$$

where ψ_T is the pdf. of a standard student-t distributed variable. \square

A.4 Proofs Bayes Decision Rules

Theorem A.4.1 (Bayes Decision Rule under Quadratic Loss). *Let $\mathcal{L}(r, d) = K(d - r)^2$ be the (quadratic) loss function. Then $\mathbb{E}(R|X)$ minimizes the posterior expected loss, eg.*

$$\arg \min_d \int_r K(d - r)^2 f_{R|X}(r; x) dr = \mathbb{E}(R|X). \quad (\text{A.4.1})$$

By definition this implies that the Bayes Decision Rule under quadratic loss is $\mathbb{E}(R|X) = \tilde{r}$.

²These parameters can be expressed in the parameters of the prior information and the test results by Equation (5.1.7).

Proof. To find the minima mentioned in Equation (A.4.1) the posterior expected loss is differentiated with respect to d and then set equal to 0. The posterior expected loss can be expressed as

$$\int_r K(d-r)^2 f_{R|X}(r; x) dr = \int_r K(d^2 - 2dr + r^2) f_{R|X}(r; x) dr. \quad (\text{A.4.2})$$

Differentiating and setting equal to zero results in

$$\begin{aligned} \frac{\partial}{\partial d} \int_r K(d^2 - 2dr + r^2) f_{R|X}(r; x) dr &= \int_r K(2d - 2r) f_{R|X}(r; x) dr \\ &= 2dK \int_r f_{R|X}(r; x) dr - 2K \int_r r f_{R|X}(r; x) dr \\ &= 2K(d - \mathbb{E}(R|X)) = 0 \end{aligned} \quad (\text{A.4.3})$$

Which gives $\tilde{r} = E(R|X)$. □

Theorem A.4.2 (Bayes Decision Rule under Bi-Linear Loss). *Let $\mathcal{L}(r, d)$ be the bi-linear loss function. The Bayes Decision Rule is then the $100 \frac{c_1}{c_1+c_2}$ -percentile estimate of the posterior distribution of R .*

Proof. To prove the theorem given above the posterior expected loss is calculated and then the minimum is found by setting the derivative of this expression to 0.

$$\begin{aligned} \mathbb{E}_{R|X}(\mathcal{L}(d, r)) &= \int_{-\infty}^{\infty} \mathcal{L}(d, r) f_{R|X}(r; x) dr \\ &= \int_{-\infty}^d c_2(d-r) f_{R|X}(r; x) dr + \int_d^{\infty} c_1(r-d) f_{R|X}(r; x) dr \\ &= c_2 d F(r) - c_2 \int_{-\infty}^d r f_{R|X}(r; x) dr + c_1 \int_d^{\infty} r f_{R|X}(r; x) dr - c_1 d [1 - F_{R|X}(r, x)] \end{aligned} \quad (\text{A.4.4})$$

The derivative of this expression to d is equal to

$$\frac{\partial}{\partial d} \mathbb{E}_{R|X}(\mathcal{L}(d, r)) = (c_1 + c_2) F_{R|X}(r; x) - c_1 \quad (\text{A.4.5})$$

Notice $\lim_{d \rightarrow \infty} \frac{\partial}{\partial d} \mathbb{E}_{R|X}(\mathcal{L}(d, r)) = c_2 \geq 0$. Thus this expectation is minimized if

$$F_{R|X}(d) = \frac{c_1}{c_1 + c_2}. \quad (\text{A.4.6})$$

Therefore the Bayes decision rule for bi-linear loss is given as

$$\tilde{r} = F_{R|X}^{-1}\left(\frac{c_1}{c_1 + c_2}\right) \triangleq r_{\frac{c_1}{c_1+c_2}} \quad (\text{A.4.7})$$

□

A.5 Direct Numerical Integration in Humidity Case

In this section a derivation of the direct numerical integration approach of the Bayesian method applied to the humidity case is given. The first step defining the prior information is exactly the same as given in subsection 8.2. The next step defining the likelihood is different then was done in section 8.2, as there the HUGIN-approach was used. Here a more mathematical procedure is followed. Still the graphical model that was used can be helpful in the derivation as it shows which transformations will be performed deriving the likelihood.

The likelihood function is deducted from the observation models given as Equations (8.2.4), (8.2.5), and (8.2.6). By definition the likelihood function is given by the following probability density function

$$f_{\Omega_M|R_v}(\omega_m; r) \quad (\text{A.5.1})$$

As given by observation model 8.2.5 all measurements at one specific time are dependent on the conversion error. Applying the LTP by conditioning on the error of conversions leads to

$$f_{\Omega_M|R_v}(\omega_m; r) = \iint f_{\Omega_M|R_v,A,B}(\omega_m; r, a, b) f_{A,B}(a, b) d(a, b). \quad (\text{A.5.2})$$

The joint density function is given by the prior information thus concentrate on $f_{\Omega_M|R_v,A,B}(\omega_m; r, a, b)$. Using the humidity relation ,Equation (8.2.6), on the conditioned variables in this function results in

$$f_{\Omega_M|R_v,A,B}(\omega_m; r, a, b) = f_{\Omega_M|\Omega_R}(\omega_m; -\frac{1}{b} \log \left(\frac{v_0}{a} - \frac{v_0 - v_c}{ar\sqrt{t_{insp}v_0}} x \right)). \quad (\text{A.5.3})$$

Now a transformation from Ω_m to ε is performed to determine the probability density function. This means using observation model, Equation (8.2.4). The Jacobian for this transformation equals

$$J = \left| \frac{\partial \varepsilon}{\partial \Omega_M} \right| = \left| \frac{1}{\Omega_R} \right|. \quad (\text{A.5.4})$$

Taking all errors being independent of each other, the transformation results in the following expression.

$$\begin{aligned} f_{\Omega_M|R_v,A,B}(\omega_m|r, a, b) &= \prod_{i=1}^n \left[f_{\varepsilon} \left(\frac{\omega_m^{(i)}}{\omega_r^{(i)}(r, a, b)} \right) \frac{1}{\omega_r^{(i)}(r, a, b)} \right] \\ &= \prod_{i=1}^n \left[f_{\varepsilon} \left(-\frac{\omega_m^{(i)} b}{\log \left(\frac{v_0}{a} - \frac{v_0 - v_c}{ar\sqrt{t_{insp}v_0}} x^{(i)} \right)} \right) - \frac{b}{\log \left(\frac{v_0}{a} - \frac{v_0 - v_c}{ar\sqrt{t_{insp}v_0}} x^{(i)} \right)} \right] \end{aligned} \quad (\text{A.5.5})$$

Where index (i) indicates at which measurement-position (depth) the measurement is taken. Notice that $\varepsilon \sim \mathcal{N}(1, 0.05)$ which implies that in principle a known expression is found for the likelihood. In principal because the integration over a,b still needs to be performed. This is thus done by direct numerical integration, which is implemented in MATLAB. The program is added in Appendix C. The next step in the Bayesian method is to calculate the posterior distribution. This is done by using Bayes' formula, e.g.

$$f_{R_v|\Omega_M}(r; \omega_m) \propto f_{\Omega_M|R_v} f_{R_v}(r) \quad (\text{A.5.6})$$

This was also done numerically in MATLAB. The resulting m-code can be used to perform the Bayesian method, to perform the update of the stochast R_v , and check if HUGIN did perform the update correct (which it did!). This was done by comparing the predictive distribution from HUGIN with the one from direct numerical integration approach.

Appendix B

Controlling HUGIN

This section contains a small report describing the use of the HUGIN API's in MATLAB.

B.1 Introduction

This report explains how HUGIN can be addressed by MATLAB and Visual Basic (for Excel) by using the HUGIN *Application Programming Interface* (API). HUGIN has now developed a Visual Basic ActiveX server which makes it possible to access the HUGIN API's from many different applications. Which was previously only supported from C-libraries.

The emphasis in this report is mainly on use of the API with MATLAB. Briefly usage from Visual Basic is discussed, this because the available help file on the ActiveX Server is based on Visual Basic. This help file¹ can be found in the HUGIN installation directory as **The HUGIN API ActiveX Server 6.3 Help**. This help file also contains some elementary examples and the API reference manual. This reference manual is needed for reference to available object properties and for reference to all possible functions.

This report can be read without any pre-knowledge of the HUGIN ActiveX server, still basic knowledge on belief nets is needed. To understand this report elementary knowledge of the Visual Basic programming language is needed (only for chapter 2) and knowledge on the use of MATLAB is needed. This report was based on usage with MATLAB 7 Release 14. If one uses another version some commands might not be executed. This is due to changes in different version of MATLAB in the function 'invoke'.

¹When in this report one is referred to 'the help file', without any reference to a specific help file this help file is intended.

B.2 Installation

Installation in Windows

During the installation of HUGIN Researcher, the HUGIN API's are placed in their own subfolder in the main installation folder. Another file is placed in the windows system32 folder, namely `nphapi63.dll`². Before this file can be used by Windows it needs to be registered with Windows. This should be done automatic during the installation but often this does not happen and therefore perform the following steps to do this manually. Open a dos-box in windows. And execute the following command `regsvr32 hapi63.dll` from `\HDE6.3X\lib`-directory which is a subdirectory of the HUGIN installation-directory. Now the server can be used in general applications, e.g. MATLAB. How this is done will be explained in chapter B.4. The next subsection will explain what to do in order to use the HUGIN API's in Visual Basic for Excel (VB).

If problems occur trying to access the HUGIN API's one should look in the HUGIN release notes to solve the problem or e-mail HUGIN at activex@hugin.com.

Library in Visual Basic

In order to use the HUGIN API's in VB the HUGIN ActiveX library needs to be added to the references of VB. This is done by selecting from the 'tools'-tab 'References'. Now a window is displayed with all the references that VB currently uses. Now tag the box with **HUGIN API ActiveX Server 6.3**. If this library is not in the list, select it via 'Browse...' from the API installation directory by selecting `hapi63.dll`. Now one can use the API's in VB, how this is done will be explained in chapter B.3.

B.3 Visual Basic for Excel

This chapter describes how to use some functions from the HUGIN API and how to change properties with the HUGIN API. Only the most basic functions are treated here by giving examples. It is meant to illustrate the syntax for using the API in VB. This section can be seen as an addition to the examples in the help file.

Small Example 1

As first the small network displayed by figure B.1 will be created with the help of VB.



Figure B.1: The Example network in HUGIN created via VB

²For some reason this is not always the case thus if this file is not found there after installing HUGIN; place this file manually there. This can be done by copying the file from the HUGIN directory

The code needed to create this network is displayed below. In the next subsections each of the commands issued will be explained.

```
Public Sub Example()
    ' Load the ActiveX Object
    Set glob = CreateObject("HAPI.Globals")
    'Create a BBN
    Set Domain = glob.GetNewDomain()
    'Create Nodes
    Set ndA = Domain.GetNewNode(hCategoryChance, hKindDiscrete)
    ndA.Label = "A"
    ndA.Name = "A"
    Set ndB = Domain.GetNewNode(hCategoryChance, hKindDiscrete)
    ndB.Label = "B"
    ndB.Name = "B"
    ' Add Structure
    Call ndA.AddParent(ndB)
    ' Save Network
    Call Domain.SaveAsNet("C:\ExampleE.net")
End Sub
```

General Commands

In order to do any work with HUGIN in VB one needs to create an object in VB to handle all operations called or performed from the API. This is done with the following command.

```
Set glob = CreateObject("HAPI.Globals")
```

Handling a Network

The following code loads a network from a HUGIN file into the ActiveX object which was created in VB by the previous code. Then this code saves the network into another HUGIN net-file.

```
Set domain=glob.LoadDomainFromNet("C:\BBN1.net", Nothing, 0)
Call domain.SaveAsNet("C:\BBN2.net")
```

The arguments 'Nothing' and '0' are additional arguments which are set to these values because they are not used here. The possibilities for using these additional arguments are explained in the API reference manual contained in the help-file.

Creating Nodes

The following code will create a node and define its type, name, and label. It is important to notice that name and label are different properties of the node. The name of a node is how HUGIN refers to the node. This implies that it must be unique for each node. According to the API tutorial the name needs to be specified in a format similar to C-identifiers. This means 'loosely' that no spaces and funny characters may be used in names. The label of a node is the name that is displayed in a node in the graphical interface.

```
Set ndA = Domain.GetNewNode(hCategoryChance, hKindDiscrete)
ndA.Label = "A"
ndA.Name = "A"
```

Other properties of a Node can be defined/alterd using set in the same way as with label and name was done. All properties available for a node can be found in the API reference manual.

Adding Structure

Once the nodes are created in the network the structure can be added. Structure is added in the visual Interface of HUGIN by drawing arcs from one node to the other. This is done from the HUGIN API with the function 'AddParent'.

```
Call ndA.AddParent(ndB)
```

Notice that the function 'AddParent' is called from the child node.

Editing States

When a discrete node is created in the network one still needs to define the discretisation. This section shows how to define/change the number of states and how to change the State Values.

```
ndA.NumberOfStates = 2
ndA.StateLabel(0) = "On"
ndA.StateLabel(1) = "Off"
```

The code displayed above changes node A to have two states; 'On', and 'Off'. The index used in the property 'StateLabel' starts at 0. Therefore the state 'On' was set with index 0. HUGIN starts every index with a 0. Notice when setting the boundaries of a node of type 'Interval' the index to define the values of the boundaries of the intervals start also at 0. The following figure shows three intervals [7, 10], [10, 13], [13, 16] at each boundary the value of the index needed in HUGIN to access this boundary is displayed above it.



Figure B.2: A plot of an interval with the index used in HUGIN

Filling Tables

The next code will display how to manually fill a (conditional) probability Table of a node.

```
ndA.Table.Data(0) = 666
```

Notice that the property 'Data' has an index attached to it. This index displays the entry in the table. The following definition gives the value of this index for each table entry.

Definition B.3.1. Let TA be the table of node A which has m columns and n rows (TA is the (conditional) probability table of node A). The index needed in the property data for element TA(i,j) is defined as

$$Index(i,j) = i - 1 + (j - 1) * (n - 1), \quad 1 \leq i \leq n, 1 \leq j \leq m. \quad (\text{B.3.1})$$

Bear in mind that HUGIN normalizes all the entries in the table such that the sum of each column equals 1, when it uses them for computation.

B.4 MATLAB

General Commands

To control HUGIN from MATLAB one has to instantiate the ActiveX server and then use the available functions in this library to alter objects created from this library. One can instantiate the HUGIN ActiveX Server into an object in MATLAB with the following command:

```
glob=actxserver('HAPI.Globals');
```

This command creates a HUGIN API object named glob, which now can be used to address functions and properties defined in the API. To see what functions are available for this object issue the command `methodsview` or `methods`, or see the reference manual. Notice that MATLAB makes a distinction between properties of an object and functions defined in the API. This distinction is apparent in the way both are called/issued.

To issue a function (or sub procedure) from the API the command `invoke` needs to be used. This command tells MATLAB which object contains the function and which function it needs to issue and with what arguments. For example if a network stored in a net-file (previously created in HUGIN e.g. `C:\ExampleE.net`) needs to be loaded into an object in MATLAB. In the reference manual the function 'LoadDomainFromNet' is found which can handle the task. The following function description comes from the reference manual.

Syntax Function

```
LoadDomainFromNet(strFileName As String, colParseMessages As
Collection, iMaxMessages As Integer) As Domain
```

This function can be issued with the next to statement.

```
domain=invoke(glob,'LoadDomainFromNet','C:\ExampleE.net',0,0);
```

The result of this statement is that the command `invoke` now tells MATLAB to issue the function `LoadDomainFromNet` defined in the object `glob`. And use as arguments for this function the filename and the default values for the last two (optional) arguments³. As a result an object is created from the API which refers to the network.

³Using the default values 0,0 means that these arguments are not used. See the reference manual for the possibilities of using these optional arguments

To edit/display a property of an object without using an API function one has to use set/get. The following command illustrates how one can edit the number of states of node A and also display it.

```
set(nodeA,'NumberOfStates',3);
a=get(nodeA,'NumberOfStates');
```

The properties of a node ⁴ e.g. number of states, children, etc. can be altered with the 'set'-command and can be retrieved from the node (object) with the 'get'-command.

Small Example2

The network displayed in figure B.1 can be created in MATLAB with the following code.

```
% Load the ActiveX Object
glob = actxserver('HAPI.Globals');
%Create the BBN
domain = invoke(glob,'GetNewDomain');
%Create Nodes
nodeA
=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete')
set(nodeA,'Label','A');
set(nodeA,'Name','A');
nodeB
=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
set(nodeB,'Label','B');
set(nodeB,'Name','B');
%Add Structure
invoke(nodeA,'AddParent',nodeB);
%Save Network
invoke(domain,'SaveAsNet','C:\ExampleM.net');
```

The following sections explains the commands issued in the code above and shows some frequently used commands. They are displayed to illustrate the syntax used in MATLAB to control HUGIN. It is again by no means meant as a complete list of all operations possible with the HUGIN API. For this see the reference manual.

Handling a Network

Creating, loading, and saving a domain is done via the following commands.

```
domain=invoke(glob,'GetNewDomain');
domain=invoke(glob,'LoadDomainFromNet',C:\ExampleM,0,0);
invoke(domain,'SaveAsNet','C:\ExampleM.net');
```

Note that when a network is created or loaded it is assigned to an object in MATLAB.

⁴In order to use these statements a variable nodeA needs to be defined in MATLAB. This variable nodeA should contain an object created from the HUGIN API referring to a node. This object needs to be created with invoke as will be shown in the next example

Table B.1: Possible types of nodes

Name	Value	Description
hCategoryChance	7	Specifies a chance node
hCategoryDecision	8	Specifies a decision node
hCategoryUtility	9	Specifies a utility node
hCategoryInstance	6	Specifies an instance node

Creating Nodes

A node is created via the following code.

```
nodeA=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
set(nodeA,'Name','A');
set(nodeA,'Label','A');
```

In order to create different kinds of nodes other arguments in this function need to be used. In the reference manual different tables can be found that show the possible arguments that can be used to create different kind of nodes. For example the following table is taken from the reference manual from the section hCategories, it displays the possible types of nodes that can be created.

There is also an table for the different subtypes in the reference manual. This defines how the states of a discrete node are defined ,e.g. labelled, boolean, numbered, or as intervals.

Adding Structure

Structure is added via the following command.

```
invoke(nodeA,'AddParent',nodeB);
```

The function 'AddParent' is called from node A with input argument node B, therefore this order in the function 'invoke' is needed. As this means that an arc is drawn from node B to node A. If the order of the nodes is changed the arc will be drawn reverse.

Editing States

The number of states and the states themselves are changed with the following code.

```
set(ndA,'NumberOfStates',2)
set(ndA,'StateLabel',0,'On')
set(ndA,'StateLabel',1,'Off')
```

Note that the index of 'StateLabel' starts at 0. HUGIN always starts indexes at 0 opposed to MATLAB which start at 1. Thus the index of state i equals i-1. See figure B.2 for the indexes used when a interval needs to be defined.

Filling Tables

To edit a table of a node manually one can use the following statement.

```
set(nodeA.Table,'Data', 0, 0.1);
```

Where the last two arguments are the 'Data' -index (see B.3.1) and the value that is entered in the table (in this case 0.1). To clarify this statement even more, for example. the matrix CPTA defined in MATLAB as

$$CPTA = \begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.9 \end{bmatrix}$$

can be copied as the table of node A with the following code in MATLAB.

```
for i=1:size(CPTA,1)
    for j=1:size(CPTA,2)
        set(nodeA.Table,'Data',i-1+(j-1)*(size(CPTA,2)-1),CPTA(i,j));
    end
end
```

As the experienced HUGIN user knows, a table of a node can also be defined by an expression or model. This is discussed in the next subsection.

Creating a Model/Expression

In HUGIN a table can be created/generated by an model/expression. While trying to create a table by a model/expression a small problem was uncovered with the API. That is there is no function available to MATLAB to create certain new objects defined by or in the API, eg. of the type HUGIN collection.⁵ This problem has been reported to HUGIN and they are working on a solution. A (temporary) solution has been found to work around this problem. This solution will be given here and the problem will be clarified by giving the example how the problem was encountered.

In order to define the table of a node by assigning an expression ie. by a formula, a model needs to be created for that node. And then insert the proper expression in this model, which of course first will have to be created. Defining a model for a node is done with the function 'NewModel' which is called from a node. This function needs as input an empty collection. This empty collection cannot be created from the API, thus a temporary dummy node (ndtemp) is created. From this ndtemp the expression for his parents (ndtemp.parents), which is empty is used to insert as an argument in the function 'NewModel'.

```
ndtemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete');
m = invoke(nodeA,'NewModel',ndtemp.parents);
```

The commands given above does the same as when selecting in HUGIN from the table menu 'Switch to Expression'. Now that the node A is capable of holding a model, the expression is set. For instance the expression is set such that the node has the same table (distribution) as node B.

```
%Create expression for node B
expr1 = invoke(glob,'GetNewNodeExpression',nodeB);
%Set expression in model
set(m,'Expression',0,expr1);
```

⁵This problem does not occur in VB because this language can create new objects with the command 'NEW'.

Note that it is only possible to set an expression as a model. Thus the statement `set(m,'Expression',0,nodeB);` will return an error, as node *B* is of node-type and not an expression.

The following code shows how to put a string as an expression. This is done with the help of the function 'GetNewExpressionFromString'.

```
%Create dummy node to create empty expression
m = invoke(nodeA,'NewModel',ndtemp.parents);
%Create expression from string
expr=invoke(domain,'GetNewExpressionFromString', 'Normal(5,4)', 0,
0);
%Set expression as model for node A
set(m,'Expression',0,expr);
```

Now node A has a normal distribution with parameters (5,4). Notice that HUGIN recognizes the operator 'Normal'. A full list of all the operators available can be found in the reference manual under Types subsection operators.

The next code shows how to create an expression using multiple operators. This is done by creating a combined expression to put in as the model from other expressions. For example the expression for node A will be defined as the product of two (parent) nodes (nodeB,nodeC).

```
%Create empty expression from dummy node
exprs = ndtemp.parents;
%Create expressions from nodes
exprB=invoke(glob,'GetNewNodeExpression',nodeB);
exprC=invoke(glob,'GetNewNodeExpression',nodeC);
%Combine node expression in one expression
invoke(exprs,'Add',exprB);
invoke(exprs,'Add',exprC);
%Create product expression
expr1=invoke(glob,'GetNewCompositeExpression','hOperatorMultiply',exprs);
%Create model for node A and set the proper expression
m =invoke(nodeA,'NewModel',ndtemp.parents);
set(m,'Expression',0,expr1);
```

Note that a dummy node `ndtemp` was used, which still needs to be deleted.

Compiling

When a network is built and the interface is created in MATLAB it can be compiled. This is the same action as pressing the 'lightning'-button in HUGIN. Notice that the network needs to be formulated correctly otherwise compiling errors occur. These will be returned to MATLAB which can be used to backtrack the problem. Compiling can be done with the following statement.

```
invoke(domain,'Compile');
```

Executing this statement can take some time.

Get Marginal Distribution

When a network is compiled one can access the marginal distributions of the nodes. In order to do so an expression of the node is needed. For example when there is an unique parent node , (node B) for node A, it can be used to get the expression (eg. use `ndB.children`). Thus getting the marginal of node A is done via the statement.

```
tableM=invoke(domain,'GetMarginal',ndB.children);
```

Notice that this command can only be issued in a compiled domain. And the returned object, here stored in variable `tableM` needs to be handled as all tables are handled in the HUGIN API (See section B.4) Note that an expressions for a node cannot be created directly in MATLAB see B.4. Therefore if there is no expression available for the node from which the marginal distribution is needed again a work around is needed. I have used the following work around. Create (before compiling the network) a child node for the node which the marginal distribution is needed. Use the expression for the parents of that node as the expression for the node to get the marginal from. Before saving the network don't forget to delete this node.

Set Evidence

Selecting evidence can only be done if the domain is compiled. When doing this from the API one has to select the state one wants to set the evidence and then propagate the findings. This is different from the graphical interface. There propagating is done automatically, in the API this is not the case. The following code shows how the second state is selected for node A. And all findings are propagated in the default way (hence the arguments `'hEquilibriumSum','hModeNormal'`).

```
invoke(ndA,'SelectState',2);
invoke(domain,'Propagate','hEquilibriumSum','hModeNormal');
```

B.5 Complete Examples MATLAB

This chapter contains four m-files, which can be copied and run in MATLAB. The first two are meant to illustrate the use of all statements discussed in the previous chapter. Notice that they create files in the root (C:\). The following network should appear. The last two can be used to create a high level toolbox in MATLAB.

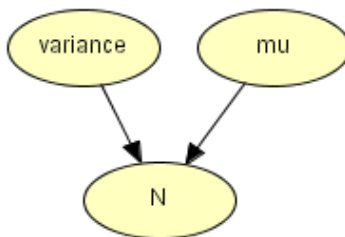


Figure B.3: The example network in HUGIN

Complete Example 1

```

%HuginExample.m
%C.J.L.Rothkrantz 2005
%Load Hugin Server
glob=actxserver('HAPI.Globals');
%Create Network
domain=invoke(glob,'GetNewDomain');
%Create Nodes mu,variance,N
ndMu=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndMu,'Name','mu'); set(ndMu,'Label','mu');
set(ndMu,'SubType','hSubtypeName'); set(ndMu,'StateValue',0,0);
ndVariance=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndVariance,'Name','variance');
set(ndVariance,'Label','variance');
set(ndVariance,'SubType','hSubtypeName');
set(ndVariance,'StateValue',0,1);
ndN=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndN,'Name','N'); set(ndN,'Label','N');
set(ndN,'SubType','hSubtypeName');
%creating discretisation for N
set(ndN,'NumberOfStates',5); A=linspace(-5,5,4); for i=1:4
    set(ndN,'StateValue',i,A(i));
end
%Add StructureCreate arcs
invoke(ndN,'AddParent',ndVariance); invoke(ndN,'AddParent',ndMu);
%Setting N normal distributed
exprMu=invoke(glob,'GetNewNodeExpression',ndMu);
exprVariance=invoke(glob,'GetNewNodeExpression',ndVariance);
ndtemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); exprs=ndtemp.parents;
invoke(exprs,'Add',exprMu); invoke(exprs,'Add',exprVariance); m =
invoke(ndN,'NewModel',ndtemp.parents);
newexp=invoke(glob,'GetNewCompositeExpression','hOperatorNormal',
exprs); set(m,'Expression',0,newexp); invoke(ndtemp,'Delete');
%Save network
invoke(domain,'SaveAsNet','C:\HuginExample.net');

```

Complete Example 2

```

%Load Hugin Server
glob=actxserver('HAPI.Globals');
%Load Network
domain=invoke(glob,'LoadDomainFromNet','C:\HuginExample.net',0,0);
%Create variables for Nodes mu,variance,N

```

```

ndMu=invoke(domain,'GetNodeByName','mu');
ndVariance=invoke(domain,'GetNodeByName','variance');
ndN=invoke(domain,'GetNodeByName','N');
%Create a second value for variance
set(ndVariance,'NumberOfStates',2)
set(ndVariance,'StateValue',1,2);
%Save network
invoke(domain,'SaveAsNet','C:\HuginExample2.net');
%Compile Network
invoke(domain,'Compile');
%Save Compiled network
invoke(domain,'SaveAsHKB','C:\HuginExample2_comp.hkb',0);
%Set evidence on variance
invoke(ndVariance,'SelectState',1);
invoke(domain,'Propagate','hEquilibriumSum','hModeNormal');
%save Case
invoke(domain,'SaveCase','C:\HuginExample2_case.hcs');

function res=NewCNode(dom,name)
%function res=NeCNode(dom,name)
%Create Chance Node with intervals
ndNew=invoke(dom,'GetNewNode','hCategoryChance','hKindDiscrete');
ndNew.Name = name; ndNew.Label=name; ndNew.SubType
='hSubtypeInterval'; res=ndNew;

function SetExpression2(glob,dom,hOperator2,node,nd1,nd2)
%function SetExpression2(glob,dom,hOperator2,node,nd1,nd2)
expr1=invoke(glob,'GetNewNodeExpression',nd1);
expr2=invoke(glob,'GetNewNodeExpression',nd2);
%Create dummy nodes
ndtemp1=invoke(dom,'GetNewNode','hCategoryChance',
'hKindDiscrete');
ndtemp2=invoke(dom,'GetNewNode','hCategoryChance',
'hKindDiscrete');
%Create temp expressions
exprs=ndtemp1.parents; invoke(exprs,'Add',expr1);
invoke(exprs,'Add',expr2); m =
invoke(node,'NewModel',ndtemp2.parents);
%Create combined expression
newexp=invoke(glob,'GetNewCompositeExpression',hOperator2, exprs);
%Set expression
set(m,'Expression',0,newexp);
%remove dummy nodes
invoke(ndtemp1,'Delete'); invoke(ndtemp2,'Delete');

```

Appendix C

Codes

This chapter contains all m-files, MATLAB codes referred to in this thesis. These m-files were written in MATLAB 7.0 release 14. Also a DPL-code is added. This code was written in the trail version of DPL 6.0.

BUAN.m

```
%% Procedure BUAN(alpha_p,beta_p,m_p,lambda_p,data_D)
%Determines the predictive distribution of R in the Timber Case.
%Applying the NIG-model. It produces a plot of the predictive
%distribution with n2 points. And a plot of the prior information with
%n3 points

%% Define Constants
%Number of points used for the plots
n2=300; n3=50;
%Prior Information
alpha_p=9; beta_p=0.01; m_p=100; lambda_p=8;
%Data
data_D=[120,110];

%% Initialization
r=linspace(50,150,n2); fr_pred=[];

%% Updating
for i=1:length(data_D)
    [alpha,beta,m,lambda]=Uppar(alpha_p,beta_p,m_p,lambda_p,data_D(1:i));
    %Predictive distribution
```

```

C1=(beta^(-alpha)*GAMMA(alpha+1/2))/(sqrt(2*pi*(1+lambda))*GAMMA(alpha));
fr_pred=cat(1,fr_pred,C1*(1/beta+((r-m).^2)/(2*(1+lambda))).^(-alpha-1/2));
end

%% Post processing
%%Plot Results
figure;
plot(r,normpdf(r,m_p,1/((alpha_p-1)*beta_p)), 'r');
hold;
title('Prior and Predictive distribution of R - Using the NIG-model');
plot(repmat(r,size(data_D'))',fr_pred');
legend('prior','predictive with 1 datapoint','predictive with 2 datapoints')
hold off;

% Plot prior info
figure; NIGPlot(alpha_p,beta_p,m_p,lambda_p,50,150,1,30,n3);

```

BUDMuAll.m

```

%% Procedure BUDMuAll
%Computes Bayes Update for Camaru by DNI and BL
data_d=[121.3144687,112.55689275,111.8634926];
data_nd=[105.0329832,102.8546259,103.9495366]; mu_m=75.29;
sigma_m=16.2; sigma=19.69088717; sigma_a=10;

%Size Data
n_d = length(data_d); n_nd = length(data_nd);

%Integration bounds
k = 4; mu_nder=mu_m-k.*sigma_m; mu_boven=mu_m+k.*sigma_m;
nmu=500; MU=linspace(mu_nder, mu_boven, nmu); dmU=MU(2)-MU(1);
nd=100; D=linspace(mu_m-4*sigma,mu_m+4*sigma_m,nd); dd=D(2)-D(1);
l = 4; a_nder=-l.*sigma_a; a_boven=l.*sigma_a; na=100;
A=linspace(a_nder, a_boven, na); da=A(2)-A(1);

%Prior for mu
prior_m = normpdf(MU,mu_m,sigma_m); tic;
%Destructive Data
if n_d>0
    flik = ones(length(MU),1);
    for i=1:n_d
        lik = (normpdf(data_d(i),MU,sigma));
    end
end

```



```

        flik = lik' .* flik;
    end
    fpost=flik'.*prior_m;
    C=sum(fpost)*dmu;
    fpostd=1/C*fpost;
else
    fpostd=prior_m
end

%Non-Destructive Data
if n_nd>0
    som=zeros(length(MU),1);
    for j=1:length(A)
        flik = ones(length(MU),1);
        for i=1:length(data_nd)
            lik = (normpdf(data_nd(i)-A(j),MU,sigma)*normpdf(A(j),0,sigma_a));
            flik = lik' .* flik;
        end
        som = som+flik;
    end
    fpost=som'.*fpostd;
    C=sum(fpost)*dmu;
    fpost=1/C*fpost;
else
    fpost=fpostd;
end; toc;

%post processing
clf; plot(MU,fpost,'b'); hold;
plot(MU,normpdf(MU,mu_m,sigma_m),'g');

%Calc posterior MU Bayes Linear
%Destructive Data
tic; CovMDi=sigma_m^2; CovMD=CovMDi*ones(1,n_d);
VarD=diag(ones(1,n_d)*sigma_m^2)+sigma_m^2*ones(n_d,n_d);

%Non Destructive Data
CovMNDi=sigma_m^2; CovMND=CovMNDi*ones(1,n_nd);
VarND=diag(ones(1,n_nd)*(sigma_a^2+sigma_m^2))+sigma_m^2*ones(n_nd,n_nd);

%Join Data
CovMAD=cat(2,CovMD,CovMND);
VarD=cat(1,VarD,sigma_m^2*ones(n_nd,n_d));
VarND=cat(1,sigma_m^2*ones(n_d,n_nd),VarND);
VarAD=cat(2,VarD,VarND); data_ad=cat(2,data_d,data_nd);

```

```

%Adjust Bayes Linear
EadM=mu_m+CovMAD/VarAD*(data_ad-mu_m)';
VadM=(sigma_m^2)-CovMAD/VarAD*CovMAD'; toc;

%post Processing
plot(MU,normpdf(MU,EadM,sqrt(VadM)),':m'); legend('posterior via
DNI','prior MU','posterior via BL');

```

HUMCreateNW.m

```

tic; domain=invoke(glob,'GetNewDomain');

%Node Rv, LogRv
ndRv=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndRv,'Name','Rv'); set(ndRv,'Label','Rv');
set(ndRv,'SubType','hSubtypeInterval');
set(ndRv,'NumberOfStates',n_disc); Rv=linspace(0,8,n_disc); for
i=1:n_disc
    set(ndRv,'StateValue',i-1,Rv(i));
end ndlogRv=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndlogRv,'Name','logRv');
set(ndlogRv,'Label','logRv');
set(ndlogRv,'SubType','hSubtypeInterval');
set(ndlogRv,'NumberOfStates',n_disc); for i=2:n_disc
    set(ndlogRv,'StateValue',i-1,log(Rv(i)));
end invoke(ndRv,'AddParent',ndlogRv);
ndtemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete');
expr1=invoke(glob,'GetNewNodeExpression',ndlogRv);
exprs=ndtemp.parents;
invoke(exprs,'Add',expr1);expr2=invoke(glob,'GetNewCompositeExpression','hOperatorExp',exprs
); m= invoke(ndRv,'NewModel',ndtemp.parents);
set(m,'Expression',0,expr2); m2=
invoke(ndlogRv,'NewModel',ndtemp.parents); expr3=
invoke(domain,'GetNewExpressionFromString',priorfunction,0, 0);
set(m2,'Expression',0,expr3);

%Create Constantes
ndV0=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndV0,'Name','V0'); set(ndV0,'Label','V0');
set(ndV0,'SubType','hSubtypeNumber');
set(ndV0,'NumberOfStates',1); set(ndV0,'StateValue',0,V0);

```

```

ndtinsp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndtinsp,'Name','tinsp');
set(ndtinsp,'Label','tinsp');
set(ndtinsp,'SubType','hSubtypeNumber');
set(ndtinsp,'NumberOfStates',1);
set(ndtinsp,'StateValue',0,tinsp);

%Create Constants
ndVc=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndVc,'Name','Vc'); set(ndVc,'Label','Vc');
set(ndVc,'SubType','hSubtypeNumber');
set(ndVc,'NumberOfStates',1); set(ndVc,'StateValue',0,Vc);

%Create Node A,B
ndA=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndA,'Name','A'); set(ndA,'Label','A');
set(ndA,'SubType','hSubtypeInterval');
set(ndA,'NumberOfStates',n_disc);
A=linspace(150.18*0.8,150.18*1.2,n_disc); for i=2:n_disc
    set(ndA,'StateValue',i-1,A(i));
end

ndB=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndB,'Name','B'); set(ndB,'Label','B');
set(ndB,'SubType','hSubtypeInterval');
set(ndB,'NumberOfStates',n_disc);
B=linspace(1.21*10(-5)*0.8,1.21*10(-5)*1.2,n_disc); for
i=2:n_disc
    set(ndB,'StateValue',i-1,B(i));
end m= invoke(ndA,'NewModel',ndtemp.parents); expr1=
invoke(domain,'GetNewExpressionFromString',Afunction,0, 0);
set(m,'Expression',0,expr1); m2=
invoke(ndB,'NewModel',ndtemp.parents); expr2=
invoke(domain,'GetNewExpressionFromString',Bfunction,0, 0);
set(m2,'Expression',0,expr2);

%Create Temp nodes for A,B
ndAtemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndAtemp,'Name','Atemp');
set(ndAtemp,'Label','Atemp'); invoke(ndAtemp,'AddParent',ndA);
ndBtemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndBtemp,'Name','Btemp');
set(ndBtemp,'Label','Btemp'); invoke(ndBtemp,'AddParent',ndB);

```

```

OmegaR=linspace(-25000,25000,n_disc);%Omega_R
%aanpassen per meting!!!
%aanpassen per meting!!!
%aanpassen per meting!!!
%aanpassen per meting!!!
OmegaM=linspace(-25000,25000,n_disc);%Omega_M
E=linspace(0.8,1.2,n_disc);%Error
%loop over observations
V=linspace(130,V0,n_disc); for j=1:n_obs
    for i=1:n
        ndVinsp=invoke(domain,'GetNewNode','hCategoryChance', 'hKindDiscrete');
        name=strcat(strcat(strcat('V_insp_',num2str(j)),'_'),num2str(i));
        set(ndVinsp,'Name',name);
        set(ndVinsp,'Label',name);
        set(ndVinsp,'SubType','hSubtypeInterval');
        set(ndVinsp,'NumberOfStates',n_disc-1);
        %discretiseer Vinsp
        for p=2:n_disc
            set(ndVinsp,'StateValue',p-1,V(p));
        end
        invoke(ndVinsp,'AddParent',ndVc);
        invoke(ndVinsp,'AddParent',ndRv);
        invoke(ndVinsp,'AddParent',ndtinsp);
        invoke(ndVinsp,'AddParent',ndV0);
        %Model ndVinsp
        exprs=ndtemp.parents;
        expr1=invoke(glob,'GetNewNodeExpression',ndVc);
        invoke(exprs,'Add',expr1);
        expr2=invoke(glob,'GetNewCompositeExpression','hOperatorNegate',exprs);
        expr3=invoke(glob,'GetNewNodeExpression',ndV0);
        exprs2=ndtemp.parents;
        invoke(exprs2,'Add',expr3);
        invoke(exprs2,'Add',expr2);
        expr4=invoke(glob,'GetNewCompositeExpression','hOperatorAdd',exprs2);
        expr5=invoke(glob,'GetNewNumberExpression',-sensor(1,j)); %afstand sensor!!!
        exprs3=ndtemp.parents;
        invoke(exprs3,'Add',expr4);
        invoke(exprs3,'Add',expr5);
        expr6=invoke(glob,'GetNewCompositeExpression','hOperatorMultiply',exprs3);
        expr7=invoke(glob,'GetNewNodeExpression',ndtinsp);
        exprs4=ndtemp.parents;
        invoke(exprs4,'Add',expr7);
        expr8=invoke(glob,'GetNewCompositeExpression','hOperatorSqrt',exprs4 );
    end
end

```

```

exprs5=ndtemp.parents;
expr9=invoke(glob,'GetNewNodeExpression',ndRv);
invoke(exprs5,'Add',expr9);
invoke(exprs5,'Add',expr8);
expr10=invoke(glob,'GetNewCompositeExpression','hOperatorMultiply',exprs5);
exprs6=ndtemp.parents;
invoke(exprs6,'Add',expr6);
invoke(exprs6,'Add',expr10);
expr11=invoke(glob,'GetNewCompositeExpression','hOperatorDivide',exprs6);
expr12=invoke(glob,'GetNewNodeExpression',ndV0);
exprs6=ndtemp.parents;
invoke(exprs6,'Add',expr12);
invoke(exprs6,'Add',expr11);
expr10=invoke(glob,'GetNewCompositeExpression','hOperatorAdd',exprs6);
m=invoke(ndVinsp,'NewModel',ndtemp.parents);
set(m,'Expression',0,expr10);
%end model

ndOmegaR=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
name=strcat(strcat('Omega_r_',num2str(j)),strcat('_',num2str(i)));
set(ndOmegaR,'Name',name);
set(ndOmegaR,'Label',name);
set(ndOmegaR,'SubType','hSubtypeInterval');
set(ndOmegaR,'NumberOfStates',n_disc);
%discretiseer OmegaR
for p=2:n_disc
    set(ndOmegaR,'StateValue',p-1,OmegaR(p));
end
invoke(ndOmegaR,'AddParent',ndVinsp);
invoke(ndOmegaR,'AddParent',ndA);
invoke(ndOmegaR,'AddParent',ndB);
%fix model
exprs=ndtemp.parents;
expr1=invoke(glob,'GetNewNodeExpression',ndVinsp);
expr2=invoke(glob,'GetNewNodeExpression',ndA);
invoke(exprs,'Add',expr1);
invoke(exprs,'Add',expr2);
expr3=invoke(glob,'GetNewCompositeExpression','hOperatorDivide',exprs);
exprs2=ndtemp.parents;
invoke(exprs2,'Add',expr3);
expr4=invoke(glob,'GetNewCompositeExpression','hOperatorLog',exprs2);
exprs3=ndtemp.parents;
expr5=invoke(glob,'GetNewNodeExpression',ndB);
invoke(exprs3,'Add',expr5);

```

```

expr6=invoke(glob,'GetNewCompositeExpression','hOperatorNegate',exprs3);
exprs4=ndtemp.parents;
invoke(exprs4,'Add',expr4);
invoke(exprs4,'Add',expr6);
expr7=invoke(glob,'GetNewCompositeExpression','hOperatorDivide',exprs4);
m=invoke(ndOmegaR,'NewModel',ndtemp.parents);
set(m,'Expression',0,expr7);
%end model

ndErr=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
name=strcat(strcat('Error_',num2str(j)),strcat('_',num2str(i)));
set(ndErr,'Name',name);
set(ndErr,'Label',name);
set(ndErr,'SubType','hSubtypeInterval');
set(ndErr,'NumberOfStates',n_disc);
%discretise Error
for p=2:n_disc
    set(ndErr,'StateValue',p-1,E(p));
end
m= invoke(ndErr,'NewModel',ndtemp.parents);
expr= invoke(domain,'GetNewExpressionFromString',Efunction,0, 0);
set(m,'Expression',0,expr);
%Create Temp nodes Err_j
ndErrtemp=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
name=strcat(strcat('ErrTemp_',num2str(j)),strcat('_',num2str(i)));
set(ndErrtemp,'Name',name);
set(ndErrtemp,'Label',name);
invoke(ndErrtemp,'AddParent',ndErr);
%END tempNode

ndOmegaM=invoke(domain,'GetNewNode','hCategoryChance','hKindDiscrete');
name=strcat(strcat('Omega_m_',num2str(j)),strcat('_',num2str(i)));
set(ndOmegaM,'Name',name);
set(ndOmegaM,'Label',name);
set(ndOmegaM,'SubType','hSubtypeInterval');
set(ndOmegaM,'NumberOfStates',n_disc);
%discretiseer OmegaM
for p=2:n_disc
    set(ndOmegaM,'StateValue',p-1,OmegaM(p));
end
invoke(ndOmegaM,'AddParent',ndOmegaR);
invoke(ndOmegaM,'AddParent',ndErr);
%model voor OmegaM
exprs=ndtemp.parents;

```

```

    expr1=invoke(glob,'GetNewNodeExpression',ndOmegaR);
    expr2=invoke(glob,'GetNewNodeExpression',ndErr);
    invoke(exprs,'Add',expr1);
    invoke(exprs,'Add',expr2);
    expr3=invoke(glob,'GetNewCompositeExpression','hOperatorMultiply',exprs);
    m=invoke(ndOmegaM,'NewModel',ndtemp.parents);
    set(m,'Expression',0,expr3);
    %end model
end
end
%insert Time still dryness
nt=13; T=linspace(0,84,nt); dT=T(3)-T(2);
ndTDry=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); name='TDry'; set(ndTDry,'Name',name);
set(ndTDry,'Label',name);
set(ndTDry,'SubType','hSubtypeInterval');
set(ndTDry,'NumberOfStates',nt+1);
%discretiseer TDry
for i=2:nt+1
    set(ndTDry,'StateValue',i-1,T(i-1));
end invoke(ndTDry,'AddParent',ndRv);
%Model TDry
exprs=ndtemp.parents;
expr1=invoke(glob,'GetNewNumberExpression',20);
expr2=invoke(glob,'GetNewNodeExpression',ndRv);
invoke(exprs,'Add',expr1); invoke(exprs,'Add',expr2);
expr3=invoke(glob,'GetNewCompositeExpression','hOperatorDivide',exprs);
exprs2=ndtemp.parents; invoke(exprs2,'Add',expr3);
invoke(exprs2,'Add',expr3);
expr4=invoke(glob,'GetNewCompositeExpression','hOperatorMultiply',exprs2);
m=invoke(ndTDry,'NewModel',ndtemp.parents);
set(m,'Expression',0,expr4);
%end model
ndTDryTemp=invoke(domain,'GetNewNode','hCategoryChance',
'hKindDiscrete'); set(ndTDryTemp,'Name','TDryTemp');
set(ndTDryTemp,'Label','TDryTemp');
invoke(ndTDryTemp,'AddParent',ndTDry);

%remove temp node for creating new collections/expressions
invoke(ndtemp,'Delete'); clear ndtemp;

%save Network
invoke(domain,'SaveAsNet','D:\HuginData\HumT.net');
%winopen D:\HuginData\HumT.net

```

```

'Network Created' toc tic;
%% Compile
tic; invoke(domain,'Compile'); 'Compiled Network' toc
invoke(domain,'SaveAsHKB','D:\HuginData\HumT.hkb',0);

```

HumFinal.m

```

%% Procedure Humidity Case
clear; DisCR=0; clock tic;
%% Create HUGIN network
priorfunction='Normal(0.5816, 0.4724 ^ 2)';
Afunction='Normal(150.18, (0.05*150.18) ^ 2)';
Bfunction='Normal(1.21*10^(-5), (0.1*1.21*10^(-5)) ^ 2)';
Efunction='Normal(1, 0.05 ^ 2)'; Vc=152; V0=155; Dc=30;
n_disc=10; %number of discretisation intervals per node
n_samp=1000; %number of samples used
sensor=linspace(3,10,3); %sensor dimensions
n=15; %number of tests proposed tests
tinsp=10; %time of inspection
n_obs=size(sensor,2); nvec=1:n; C_T=45; C_S=100;

%% Load Hugin Server
glob=actxserver('HAPI.Globals');
%CreateCompNW
domain=invoke(glob,'LoadDomainFromHKB','D:\HuginData\HumT.hkb');
'Network Loaded'
%% Create samples for measurements
%Sample Test Results
logRvS=normrnd(0.5816, 0.4724,1,n_samp); RvS=exp(logRvS);
AS= repmat(normrnd(150.18,
(0.05*150.18),1,n_samp),[length(sensor),1,n]);
BS= repmat(normrnd(1.21*10^(-5),
(0.1*1.21*10^(-5)),1,n_samp),[length(sensor),1,n]);
VS=V0-( repmat(((V0-Vc)/sqrt(tinsp))./RvS),[length(sensor),1,n])).* repmat(sensor',[1,n_samp,n]);
ErrorS=normrnd(1, 0.05,length(sensor),n_samp,n);
OmegaRS=log(VS./AS)./BS; OmegaMS=OmegaRS.*ErrorS; 'Samples
Created' nt=13;
%Define Loss function
T=linspace(0,84,nt); dT=T(3)-T(2);
OmegaM=linspace(-25000,25000,n_disc);
ndTDryTemp=invoke(domain,'GetNodeByName','TDryTemp');
tableT=invoke(domain,'GetMarginal',ndTDryTemp.parents); for

```



```

p=1:nt+1
    TDry(1,p)=get(tableT,'Data',p-1);
end TDry=cumsum(TDry);
TDryPrior=interp1(TDry,0:(size(TDry,2)-1),0.8,'nearest');

Loss=zeros(1,size(T,2)); for p=1:(TDryPrior-1)
    Loss(1,TDryPrior-p)=-p*500;
end

%Perform update!
TDryS=[];
%set evidence loop over samples, loop over testnr loop over sensor
tic; for k=1:10 for j=1:n
    for i=1:n_obs
        name=strcat(strcat(strcat('Omega_m_',num2str(i)),'_'),num2str(j));
        ndMeting=invoke(domain,'GetNodeByName',name);
        %Selecte state sensor
        state=interp1([-inf,OmegaM,inf],0:(size(OmegaM,2)+1),OmegaMS(i,k,j),'nearest');
        invoke(ndMeting,'SelectState',state-1);
    end %loop over sensor
    invoke(domain,'Propagate','hEquilibriumSum','hModeNormal');
    %get marginal for TDry
    tableT=invoke(domain,'GetMarginal',ndTDryTemp.parents);
    TDry=[];
    for p=1:nt+1
        TDry(1,p)=get(tableT,'Data',p-1);
    end
    %save TDRY 80
    TDry=cumsum(TDry);
    %Create distinct version of TDry
    if (TDry(2)-TDry(1)<0.0001)
        TDryS(j,k)=TDryPrior;
        DisCR=DisCR+1;
    end
    if (TDry(end)-TDry(end-1)<0.0001)
        TDryS(j,k)=TDryPrior;
        DisCR=DisCR+1;
    else
        TDryS(j,k)=interp1(TDry,0:(size(TDry,2)-1),0.8,'nearest');
    end
end %loop over tests
invoke(domain,'RetractFindings') end 'Time for 10 runs' toc tic;
for k=11:n_samp for j=1:n
    for i=1:n_obs

```

```

    name=strcat(strcat(strcat('Omega_m_',num2str(i)),'_'),num2str(j));
    ndMeting=invoke(domain,'GetNodeByName',name);
    %Selecte state sensor
    state=interp1([-inf,OmegaM,inf],0:(size(OmegaM,2)+1),OmegaMS(i,k,j),'nearest');
    invoke(ndMeting,'SelectState',state-1);
end %loop over sensor
invoke(domain,'Propagate','hEquilibriumSum','hModeNormal');
%get marginal for TDry
tableT=invoke(domain,'GetMarginal',ndTDryTemp.parents);
TDry=[];
for p=1:nt+1
    TDry(1,p)=get(tableT,'Data',p-1);
end
%save TDRY 80
TDry=cumsum(TDry);
%distinct maken
q=0;
TDrydiff=diff(TDry);
if size(TDrydiff,2)>0
    TDry=[TDry(TDrydiff>0.001),1];
    DisCR=DisCR+1;
    while TDrydiff(q+1)<0.001
        q=q+1;
    end
end
TDryS(j,k)=interp1(TDry,q:q+(size(TDry,2)-1),0.8,'nearest');
end %loop over tests
invoke(domain,'RetractFindings')
end %loop over samples
toc;

%Calculate Expected Loss
TCount=histc(TDryS',0:(size(T,2)-1));
ProbT=(TCount(1:end,:)/n_samp); %row indicatetestes testnumber, column days
ELOSST=ProbT'*Loss'+C_T*nvec'+C_S; ELOSST=[0;ELOSST];
nvec=[0,nvec]; plot(nvec,ELOSST,'-*')
%PLAY SOUND FROM PC SPEAKER TO INDICATE END
load handel sound(y,Fs)

```

NewCNode.m

```

function res=NewCNode(dom,name)
%function res=NeCNode(dom,name)

```

```

ndNew=invoke(dom,'GetNewNode','hCategoryChance','hKindDiscrete');
ndNew.Name = name;
ndNew.Label=name;
ndNew.SubType='hSubtypeInterval';
res=ndNew;

```

NewNumNode.m

```

function res=NewNumNode(dom,name)
%function res=NewNumNode(dom,name)
ndNew=invoke(dom,'GetNewNode','hCategoryChance','hKindDiscrete');
ndNew.Name = name;
ndNew.Label=name;
ndNew.SubType
='hSubtypeNumber';
res=ndNew;

```

Nigpdf.m

```

function res=nigpdf(x,y,alpha,beta,m,lambda)
%function res=nigpdf(x,y,alpha,beta,m,lambda)
%2D-probability density function for the Normal Inverted Gamma Distribution
%x,y are variables vectors
%alpha is shape parameter beta is scale parameter for
%
%C.J.L.Rothkrantz jan 2005

[X,Y]=meshgrid(x,y); res=1./(sqrt(2*pi*lambda).*GAMMA(alpha))*Y
.^(-alpha-3/2).*beta.^(-alpha).*exp(-1/2.*((X-m).^2./(Y.*lambda))-1./(Y.*beta));

```

NIGPlot.m

```

function NIGPlot(alpha,beta,m,lambda,x_o,x_b,y_o,y_b,n)
%function NIGPlot(alpha,beta,m,lambda,x_o,x_b,y_o,y_b,n)
%
%C.J.L.Rothkrantz jan 2005
%
%Dependencies: NIGPDF.m

x=linspace(x_o,x_b,n); y=linspace(y_o,y_b,n);
mesh(x,y,NIGPDF(x,y,alpha,beta,m,lambda));

```

Numnorm.m

ENTER CODE

NumbTimbN.m

```

%% Procedure Direct Numerical Intergration applied to normal model
%Define constants
n=100; %number of points used in discretisation
%Prior information
mu_m=75.29; sigma_m=23.32;
mu_s=16.2; sigma_s=4.36;
mu_a=1; sigma_a=0.22;
%Data
data_D=[ 121.3145 112.5569 111.8635 98.7846 158.2367];
data_U=[105.0330 102.8546 103.9495 94.0723 104.8491];
%Define discretisation
mu=linspace(0,200,n); dm=mu(2)-mu(1); r=mu; dr=dm;
sigma=linspace(0.001,40,n); dsigma=sigma(2)-sigma(1);
[MU,SIGMA]=meshgrid(mu,sigma); a=linspace(0.01,2,40);
da=a(2)-a(1);
%Process Prior information
prior1=normpdf(MU,mu_m,sigma_m).*normpdf(SIGMA,mu_s,sigma_s);
prior2=1./SIGMA; prior=prior2;
%% Computation
%Destructive data
fpost_D=prior; if size(data_D,2)>0
    L_D=1;
    for i=1:size(data_D,2)
        L_D=L_D.*normpdf(data_D(i),MU,SIGMA);
    end
    fpost_D=L_D.*fpost_D;
%Determine posterior destructive data
C1=sum(sum(fpost_D).*dm).*dsigma;
fpost_D=1/C1*fpost_D;
end

%Non-destructive data
if size(data_U,2)>0
    [MU2, SIGMA2, A] = meshgrid(mu, sigma, a);
    L_U=normpdf(A, mu_a, sigma_a);
    for i=1:size(data_U,2)
        L_U = L_U.*normpdf(data_U(i)./A, MU2, SIGMA2)./A; %Notice L_U 3D matrix
    end
end

```

```

L_U=sum(L_U,3).*da;
fpost=L_U.*fpost_D;
C2=sum(sum(fpost).*dmu).*dsigma;
fpost=1/C2*fpost;
else
    fpost=fpost_D;
end

%Determine predictive distribution
fpred = []; for i=1:size(r,2)
    tempr= normpdf(r(i),MU,SIGMA);
    predrt = tempr .* fpost;
    C3= sum(sum(predrt).*dmu).*dsigma;
    fpred=cat(2,fpred,C3);
end C4=sum(fpred).*dr; fpred=1/C4*fpred; plot(r,fpred,'r')

```

PlotHumPrior.m

```

function [f1,f2]=PlotHumPrior(tinsp,Dc)
%% Sample Size:
nmax=1000000; %number of samples at each time step
nstap=50; %number of time steps
%% Time scale
teind=30; start=teind/nstap;
t=linspace(start,teind,nstap); %time
%% Parameters distribution
[A,B]=lognpar(2,1);
%% Initialise
Y=[]; h=waitbar(0,'Calculations for Plotting Reliability with
Prior Information. Please wait...');
X=normrnd(A,B,nmax,1); for
i=1:nstap
    D=exp(X)*sqrt(t(1,i)); %sample D(t(1,i))
    Y=cat(1,Y,size(nonzeros(D>Dc),1)/nmax);
    waitbar(i/nstap,h)
end close(h)
%% Plot results
C=norminv(1-Y); %reliabilty index
f1=figure; hold on;
ninsp=floor((tinsp)/teind)*nstap;
t1=t(1:ninsp);
t2=t(ninsp:size(t,2));
C1=C(1:ninsp);

```

```

C2=C(ninsp:size(t,2));
plot(t1,C1,'-r')
plot(t2,C2,':r')
xlabel('Time [days]');
ylabel('Beta [-]');
f2=figure;
hold on;
C3=linspace(0,8,100);
plot(C3,logncdf(C3,A,B),'r')

```

PlotHumXY.m

```

function [f1,f2] = PlotHumXY(f1,f2,Dc,X,FX,YRv,tinsp,tninsp,color)
%Case Hum Sample Rv
%input X,FX, give distribution for Rv updated at tinsp
%color contains color of plot this must be a string!
%X domain, FX cdf, YRv pdf
%input X,FX, give distribution for Rv updated at tinsp
%color contains color of plot this must be a string!

%Create distinct versions of X,FX
X_2=[]; FX_2=[]; j=1; for i=1:size(FX,2)-1
    if FX(1,i)~=FX(1,i+1)
        X_2(1,j)=X(1,i);
        FX_2(1,j)=FX(1,i);
        j=j+1;
    end
end X_2(1,j)=X(1,size(FX,2)); FX_2(1,j)=FX(1,size(FX,2));

if (X_2(1)~=0) & (FX_2~=0)
    X_2=[0,X_2];
    FX_2=[0,FX_2];
end

%Sample Size:
nmax=1000000; %number of samples at each time step
nstap=20; %number of time steps
%Time scale
teind=30;
if tninsp==666
    tninsp=teind;
end
t=linspace(tinsp,teind,nstap);%time

```

```

%initialise
h=waitbar(0, strcat(strcat('Computation for Plotting Reliability
after update at t= ', num2str(tinsp)), ' in Progress. Please
wait...'));
Y=[];
U=rand(nmax,1);
R=interp1(FX_2,X_2,U);%random samples from distribution FX
for i=1:nstap
    D=R*sqrt(t(1,i)); %sample D(t(1,i))
    Y=cat(1,Y,size(nonzeros(D>Dc),1)/nmax);
    waitbar(i/(nstap),h)
end close(h)
%Plot results
figure(f1);
C=norminv(1-Y);%reliabilty index
ntninsp=ceil((tninsp-tinsp)/(teind-tinsp)*nstap); t1=t(1:ntninsp);
C1=C(1:ntninsp); plot(t1,C1, strcat('-',color)); if ntninsp~=nstap
    t2=t(ntninsp:size(t,2));
    C2=C(ntninsp:size(C,1));
    plot(t2,C2, strcat(':',color));
end figure(f2); plot(X_2,FX_2, strcat('-',color));

```

IGPdf.m

```

function res =igpdf(y,alpha,beta)
%function res =igpdf(y,alpha,beta)
%probability density function for the Inverted Gamma Distribution
%alpha: shape parameter beta: scale parameter
%
%C.J.L.Rothkrantz jan 2005
%
res=1/GAMMA(alpha)*y.^(-alpha-1)*beta^(-alpha).*exp(-1./(y*beta));

```

IGPlot.m

```

function IGPlot(alpha,beta,n,y_o,y_b)
%function IGPlot(alpha,beta,n,y_o,y_b)
%alpha:scale beta: shape n: number of points
%y_o is lowerbound for plot
%y_b is upperbound for plot
%
%C.J.L.Rothkrantz jan 2005
%

```

```

%Dependencies IGPDF.m
%
y=linspace(y_o,y_b,n);
plot(y,IGPDF(y,alpha,beta));

```

SampleME8.m

```

%%function sampling mu,eta calculatin expected loss
%% Initialisation
%clc;
%clf;
hold on; h=waitbar(0,'Calculations in Progress. Please wait...');

%Define constants
n_samp=50000; %number of samples
n=20;%number of tests maxed by batchsize
ni=10;%numner of runs

%Prior information
m=80; lambda=8; alpha=8; beta=0.0005;
KARRV=[0,30,35,40,50,60,70];%Classification Characteristic Values
price_p=500; batch_size=100;
%Processing Prior Information
nvec=(1:1:n);
KARRprior=m+sqrt((1+lambda)/(beta*alpha))*tinv(0.05,2*alpha);
%Characteristic Value Prior
nKVprior=max(histc(KARRprior,[KARRV,inf])*(1:length(KARRV)+1)',1);
%Determining Loss function
Loss=[1]; if nKVprior==1
    Loss=[0];
end for i=2:size(KARRV,2)
    Loss=cat(2,Loss,-1+1.1^(nKVprior-i));
end Loss=Loss*price_p*batch_size;
C_T=125;%costs of performing tests
ELOSS=[];

%% Computation
lambdaS=lambda./repmat((ones(1,size(nvec,2))+nvec*lambda)',1,n_samp);
alphaV=alpha+1/2*nvec'; alphaS=repmat(alphaV,1,n_samp);
vectinv=tinv(0.05,2*alphaV); mattinv=repmat(vectinv,1,n_samp); for
i=1:ni
    waitbar(i/(ni+1),h,ni-i+1)
    %create samples for eta (=sigma^2) from IG-distribution (alpha, beta)

```



```

ETAS=1./gamrnd(alpha,beta,n_samp,1);
%create samples for mu from normal distribution(m,lambda eta) given
%eta!
MUS=m+sqrt(lambda.*ETAS).*randn(n_samp,1);
%Create Destructive Sample Matrix fixed mu, sigma in each colum, row
%represents number of test
DS= repmat(MUS',n,1)+repmat(sqrt(ETAS'),n,1).*randn(n,n_samp);
clear ETAS MUS
sumDS=cumsum(DS,1);
sumSDS=cumsum(DS.^2,1);
%Calculate updated parameters with Destructive tests using NIG-model
mS=(m+lambda.*sumDS)./repmat((ones(1,size(nvec,2))+nvec*lambda)',1,n_samp);
betaS=repmat(2*beta*(1+lambda*nvec'),1,n_samp)./(-beta*(lambda*sumDS.^2+2*m*sumDS-m^2
*repmat(nvec',1,n_samp)-(1+lambda*repmat(nvec',1,n_samp)).*sumSDS)
+repmat(2*(ones(1,size(nvec,2))+nvec*lambda)',1,n_samp));
clear DS sumDs sumSDS
%Sample Characteristic Value
KARRS=mS+sqrt((1+lambdaS)./(betaS.*alphaS)).*mattinv;
%clear mS betaS
CountT=histc(KARRS',[-inf,KARRV(2:end),inf]);
CountT=CountT(1:end-1,:);
ProbC=(CountT/n_samp)';
ELOSST=ProbC*Loss'+C_T*nvec';
ELOSST=[0;ELOSST];
ELOSS=cat(2,ELOSST,ELOSS);
clear ProbC ELOSST
end
%% Postprocessing
waitbar((ni+1)/(ni+1),h,'Plotting') plot([0,nvec]',ELOSS) close(h)

```

SampleMEHugin4.m

```

%% Procedure sampling mu,eta calculating expected loss Using HUGIN for predictive distribution

%% Initialisation
clc; clock
%h=waitbar(0,'Calculations in Progress. Please wait...');

%Define constants
n_samp=10000; %number of samples
n=20;%number of tests
ni=5;%number of runs

```

```

m=80; lambda=8; alpha=8; beta=0.0005;
KARRV=[0,30,35,40,50,60,70];%Classification Characteristic Values
price_p=500; batch_size=100;

%% Processing Prior Information
nvec=(1:1:n);
KARRprior=m+sqrt((1+lambda)/(beta*alpha))*tinv(0.05,2*alpha);%Characteristic Value Prior
nKVprior=max(histc(KARRprior,[KARRV,inf])*(1:length(KARRV)+1)',1);
%Determining Loss function
Loss=[1]; if nKVprior==1
    Loss=[0];
end for i=2:size(KARRV,2)
    Loss=cat(2,Loss,-1+1.1^(nKVprior-i));
end Loss=Loss*price_p*batch_size;
C_T=125;%costs of performing tests
ELOSS=[];

%load Hugin Server
glob=actxserver('HAPI.Globals');

%% Create Network
domain=invoke(glob,'GetNewDomain'); ndM=NewNumNode(domain,'M');
ndLambda=NewNumNode(domain,'Lambda');
ndAlpha=NewNumNode(domain,'Alpha');
ndBeta=NewNumNode(domain,'Beta'); ndMu=NewCNode(domain,'Mu');
ndInvH=NewCNode(domain,'InvH'); ndH=NewCNode(domain,'H');
ndR=NewCNode(domain,'R'); ndTemp=NewCNode(domain,'Temp');
ndRTemp=NewNumNode(domain,'RTemp');
invoke(ndRTemp,'AddParent',ndR); invoke(ndMu,'AddParent',ndM);
invoke(ndMu,'AddParent',ndLambda); invoke(ndMu,'AddParent',ndH);
invoke(ndInvH,'AddParent',ndAlpha);
invoke(ndInvH,'AddParent',ndBeta); invoke(ndH,'AddParent',ndInvH);
invoke(ndR,'AddParent',ndH); invoke(ndR,'AddParent',ndMu);
%define parameters
set(ndM,'StateValue',0,m); set(ndLambda,'StateValue',0,lambda);
set(ndAlpha,'StateValue',0,alpha);
set(ndBeta,'StateValue',0,beta);
%define discretisations
%node H en InvH
n_hug=10;%number of states
HVec=linspace(0.001,400,n_hug); MuVec=linspace(0,200,n_hug);
set(ndH,'NumberOfStates',n_hug);
set(ndInvH,'NumberOfStates',n_hug);
set(ndMu,'NumberOfStates',n_hug); for i=1:n_hug-1

```

```

    set(ndH, 'StateValue', i, HVec(i));
    set(ndInvH, 'StateValue', n_hug-i, 1/HVec(i));
    set(ndMu, 'StateValue', i, MuVec(i));
end
%node R
set(ndR, 'NumberOfStates', size(KARRV, 2)+1); for i=1:size(KARRV, 2)
    set(ndR, 'StateValue', i, KARRV(i));
end
%destructive Nodes
DVec=linspace(60,140,n_hug); %equidistant discretisation
dDVec=DVec(2)-DVec(1); for j=1:n
    name=strcat('D', num2str(j));
    nodeD=NewCNode(domain, name);
    invoke(nodeD, 'AddParent', ndMu);
    invoke(nodeD, 'AddParent', ndH);
    set(nodeD, 'NumberOfStates', n_hug);
    for i=1:n_hug-1
        set(nodeD, 'StateValue', i, DVec(i));
    end
end
%define models
%Node InvH
exprs=ndTemp.parents;
exprAlpha=invoke(glob, 'GetNewNodeExpression', ndAlpha);
exprBeta=invoke(glob, 'GetNewNodeExpression', ndBeta);
invoke(exprs, 'Add', exprAlpha); invoke(exprs, 'Add', exprBeta);
expr3=invoke(glob, 'GetNewCompositeExpression', 'hOperatorGamma', exprs);
model=invoke(ndInvH, 'NewModel', ndTemp.parents);
set(model, 'Expression', 0, expr3);
%node H
exprs=ndTemp.parents;
exprInvH=invoke(glob, 'GetNewNodeExpression', ndInvH);
expr1=invoke(glob, 'GetNewNumberExpression', 1);
invoke(exprs, 'Add', expr1); invoke(exprs, 'Add', exprInvH);
expr3=invoke(glob, 'GetNewCompositeExpression', 'hOperatorDivide', exprs);
model=invoke(ndH, 'NewModel', ndTemp.parents);
set(model, 'Expression', 0, expr3);
%node Mu
exprs=ndTemp.parents; exprs2=ndTemp.parents;
exprM=invoke(glob, 'GetNewNodeExpression', ndM);
exprLambda=invoke(glob, 'GetNewNodeExpression', ndLambda);
exprH=invoke(glob, 'GetNewNodeExpression', ndH);
invoke(exprs, 'Add', exprH); invoke(exprs, 'Add', exprLambda);
expr2=invoke(glob, 'GetNewCompositeExpression', 'hOperatorMultiply', exprs);

```

```

invoke(exprs2,'Add',exprM); invoke(exprs2,'Add',expr2);
expr3=invoke(glob,'GetNewCompositeExpression','hOperatorNormal',exprs2);
model=invoke(ndMu,'NewModel',ndTemp.parents);
set(model,'Expression',0,expr3);
%nodeR
exprs=ndTemp.parents;
exprMu=invoke(glob,'GetNewNodeExpression',ndMu);
invoke(exprs,'Add',exprMu); invoke(exprs,'Add',exprH);
expr3=invoke(glob,'GetNewCompositeExpression','hOperatorNormal',exprs);
model=invoke(ndR,'NewModel',ndTemp.parents);
set(model,'Expression',0,expr3);
%nodeD Multiple!!!
for j=1:n
    name=strcat('D',num2str(j));
    nodeD=invoke(domain,'GetNodeByName',name);
    model=invoke(nodeD,'NewModel',ndTemp.parents);
    set(model,'Expression',0,expr3);
end
%save Network
invoke(domain,'SaveAsNet','C:\SampleTimb3.net');

%% Computation
%compile
invoke(domain,'Compile'); for i=1:ni
    tic;
    %waitbar(i/(ni+1),h,ni-i+1)
    %create samples for eta (=sigma^2) from IG-distribution (alpha, beta)
    ETAS=1./gamrnd(alpha,beta,n_samp,1);
    %create samples for mu from normal distribution(m,lambda eta) given
    %eta!
    MUS=m+sqrt(lambda.*ETAS).*randn(n_samp,1);
    %Create Destructive Sample Matrix fixed mu, sigma in each colum, row
    %represents number of test
    DS=repmat(MUS',n,1)+repmat(sqrt(ETAS'),n,1).*randn(n,n_samp);
    clear ETAS MUS
    nDS=ceil((DS-DVec(1))/dDVec);
    nDS=min(max(0,nDS),n_hug-1);
    %Each Column of nDS loop!
    KARRS=[];
    for j=1:n_samp
        for i=1:n
            name=strcat('D',num2str(i));
            nodeD=invoke(domain,'GetNodeByName',name);
            invoke(nodeD,'SelectState',nDS(i,j));

```

```

        invoke(domain,'Propagate','hEquilibriumSum','hModeNormal');
    %Get KARRS
    GO=1;
    k=0;
    som=0;
    tabel=invoke(domain,'GetMarginal',ndRTemp.parents);
    while ((k<size(KARRV,2))& GO)
        k=k+1;
        tempsom=get(tabel,'Data',k);
        som=som+tempsom;
        if som >=0.05
            GO=0;
            KARRS(i,j)=k;
        end
    end
    end
    invoke(domain,'RetractFindings')
end

TCount=histc(KARRS',[-inf,(2:1:size(KARRV,2)+1)]);
ProbC=(TCount(1:end-1,:)/n_samp)'; %row indicatetestes testnumber, column strength-clas
ELOSST=ProbC*Loss'+C_T*nvec';
ELOSST=[0;ELOSST];
ELOSS=cat(2,ELOSST,ELOSS);
toc;
end

%% Postprocessing
%waitbar((ni+1)/(ni+1),h,'Plotting')
plot([0,nvec'],'ELOSS)
%close(h)

```

Uppar.m

```

function
[Nalpha,Nbeta,Nm,Nlambda]=Uppar(alpha,beta,m,lambda,data_D)
%function [Nalpha,Nbeta,Nm,Nlambda]=Uppar(alpha,beta,m,lambda,d)
%updates the parameters of the NIG-model using destructive data

n=length(data_D); Nlambda=lambda/(1+n*lambda);
Nm=(m+lambda*sum(data_D))/(1+n*lambda); Nalpha=alpha+n/2;
Nbeta=(2*beta*(1+n*lambda))/(-beta*(lambda*(sum(data_D))^2+
2*m*sum(data_D)-n*m^2-(1+n*lambda)*(sum(data_D.*data_D)))+2*(1+n*lambda));

```

DPL-code

This section shows a code used in DPL to program a decision problem in the simplified timber case. It is only shown here to demonstrate how to program code used in DPL.

```

const cost = 3;

#options -e decision
Aantal_testen.{No_tests,One_tests,Two_tests,Three_tests,Four_tests}=
  0,                // Aantal_testen.No_tests
  -cost,            // Aantal_testen.One_tests
  -cost*2,          // Aantal_testen.Two_tests
  -cost*3,          // Aantal_testen.Three_tests
  -cost*4,          // Aantal_testen.Four_tests

chance Test1.{1}=; value Test1=DPLMC.normal(100,sqrt(200));
chance Test2.{1}=; value
Test2=DPLMC.normal((10^2*Test1+1000)/(200),sqrt(50+100));
chance Test3.{1}=; value
Test3=DPLMC.normal((20^2*(Test1+Test2)+1000)/(300),sqrt((100/3)+100));
chance Test4.{1}=; value
Test4=DPLMC.normal((30^2*(Test1+Test2+Test3)+1000)/(400),sqrt(100/4+100));
chance Test5.{1}=; value
Test5=DPLMC.normal((40^2*(Test1+Test2+Test3+Test4)+1000)/(500),sqrt(100/5+100));

chance Posterior_Distribution.{1}=; value
Posterior_Distribution|Aantal_testen=
  DPLMC.normal(100,sqrt(200)),    // Aantal_testen.No_tests
  DPLMC.normal((10^2*Test1+10000)/(200),sqrt(50+100)),    // Aantal_testen.One_tests
  DPLMC.normal((100*(Test1+Test2)+10000)/(300),sqrt(100/3+100)),    // Aantal_testen.Two_tests
  DPLMC.normal((200*(Test1+Test2+Test3)+10000)/(400),sqrt(100/4+100)),    // Aantal_testen.Three_tests
  DPLMC.normal((300*(Test1+Test2+Test3+Test4)+10000)/(500),sqrt(100/5+100)),    // Aantal_testen.Four_tests

value Expected_Gain|Aantal_testen=
  -((Posterior_Distribution-100)^2)+Aantal_testen,    // Aantal_testen.No_tests
  -((Posterior_Distribution-(10^2*Test1+10000)/(200))^2)+Aantal_testen,    // Aantal_testen.One_tests
  -((Posterior_Distribution-(100*(Test1+Test2)+10000)/(300))^2)+Aantal_testen,    // Aantal_testen.Two_tests
  -((Posterior_Distribution-(200*(Test1+Test2+Test3)+10000)/(400))^2)+Aantal_testen,    // Aantal_testen.Three_tests
  -((Posterior_Distribution-(300*(Test1+Test2+Test3+Test4)+10000)/(500))^2)+Aantal_testen,    // Aantal_testen.Four_tests

sequence:
decide
  to Aantal_testen.No_tests then
    gamble on Posterior_Distribution and get Expected_Gain
  to Aantal_testen.One_tests then

```

```
    gamble on Test1 then
      gamble on Posterior_Distribution and get Expected_Gain
to Aantal_testen.Two_tests then
    gamble on Test1 then
    gamble on Test2 then
    gamble on Posterior_Distribution and get Expected_Gain
to Aantal_testen.Three_tests then
    gamble on Test1 then
    gamble on Test2 then
    gamble on Test3 then
    gamble on Posterior_Distribution and get Expected_Gain
to Aantal_testen.Four_tests then
    gamble on Test1 then
    gamble on Test2 then
    gamble on Test3 then
    gamble on Test4 then
    gamble on Posterior_Distribution and get Expected_Gain
```


Appendix D

Notation

The notation used through out this thesis has been kept as close as possible to the notational conventions commonly used in mathematics. This means that upper case letters are used for stochastic variables, and lower case letters for variables, and samples. Vectors are denoted with an underline and matrices are also denoted with uppercase letters. Elements or multiples are denoted with a index. The following list gives a glossary of the mathematical operators, the probabilistic and decision theoretic notation, and the variables used throughout this thesis. This list is only meant as fast reference, all variables are explained when they are first introduced in the text.

Mathematical Operators

$1_{A>B}$	the indicator function, i.e. equals 1 if $A > B$ and 0 if $A < B$.
$Cov(X, Y)$	the covariance (matrix) of X and Y .
$\mathbb{E}(X)$	expectation of X , given by $\mathbb{E}(X) = \int_{-\infty}^{\infty} x f_X(x) dx = \sum_{i=1}^{\infty} x_i f_X(x_i)$
$\mathbb{E}_R(\Theta R)$	expectation with respect to the distribution of R indicated by the sub-index R .
J	the Jacobian determinant, $ \frac{\partial(y_1, \dots, y_m)}{\partial(x_1, \dots, x_n)} $.
$Var(X)$	variance (matrix) of X , given by $Var(X) = \mathbb{E}((X - \mu)^2)$, where $\mu = \mathbb{E}(X)$.

Probabilistic Notation

\sim	distributed as, e.g. $X \sim \mathcal{N}(\mu, \sigma)$ means X has a normal distribution with parameters μ , and σ .
$ $	conditional symbol, e.g. $X Y$ means X given (conditional on) Y .
φ	the probability density function of the standard normal distribution ($\mu = 0, \sigma = 1$).
Φ	cummulative distribution function of the standard normal distribution ($\mu = 0, \sigma = 1$).
$f_X(x)$	the probability density function of X , $\int f_X(x)dx = F_X(x)$.
$F_X(x)$	the cumulative distribution function of X , also called the probability distribution function of X , $F_X(x) = P(X \leq x)$.
$\mathcal{G}(\alpha, \beta)$	gamma distribution with shape parameters α and scale parameter β .
$L(X)$	the likelihood (function) of X , $L(X) = f_{D X}(\delta; x)$.
$\mathcal{N}(\mu, \sigma)$	normal distribution with parameters μ , and σ .
$\mathcal{NIG}(\alpha, \beta, m, \Lambda)$	normal inverted gamma distribution , this is a joint distribution, e.g. $[M, H] \sim \mathcal{NIG}(\alpha, \beta, m, \Lambda)$ then $H \sim \mathcal{IG}(\alpha, \beta)$ and $[M H] \sim \mathcal{N}(m, \sqrt{\Lambda\eta})$.
$\mathcal{IG}(\alpha, \beta)$	inverted gamma distribution with shape parameter α , and scale parameter $\frac{1}{\beta}$.

Notation Decision Theory

\mathcal{C}	the set of all consequences.
C_T	cost of performing a single test.
C_S	(fixed) set up cost for testing.
f	an act.
\mathcal{F}	the set of all acts, i.e., $\mathcal{F} = \{f f : \mathcal{S} \rightarrow \mathcal{C}\}$.
$\mathcal{L}(d, r)$	the decision loss function describing the loss in deciding to use parameter d when the true parameter is r .
$\mathcal{L}^F(d, r, n)$	the loss function including the cost of performing n tests.
\tilde{n}	the number of tests minimizing the posterior expected risk (need not be integer!).
\tilde{r}_n	the Bayes decision rule for R for fixed n .
\mathcal{S}	the set of all states of the world.
$U(c)$	utility function, i.e., $U : \mathcal{C} \rightarrow \mathbb{R}$.
$v(f)$	expected utility of performing act f .

Glossary of Variables

$A(a), B(b)$	conversion errors.
β	reliability index defined as, $\beta = \Phi^{-1}(1 - P_f)$.
$\varepsilon(a)$	error-term in the measurement.
$\Omega_m^i(\omega_m^i)$	measured specific resistance at depth i .
$\Omega_r^i(\omega_r^i)$	actual (real) specific resistance at depth i .
$\Sigma(\sigma)$	the standard deviation of the timber.
$\Upsilon(v)$	results of a non-destructive strength test.
$D(\delta)$	results of a destructive strength test.
d_{Wall}	the average distance the humidity from within the wall needs to travel to reach the stucco.
$D(t)$	the distance in the structure where the humidity is at the critical level.
$M(\mu)$	the mean strength of the timber R .
P_f	probability of failure.
R^{Char}	the actual characteristic strength.
R_G	the graded characteristic strength.
$R(r)$	the strength of the timber.
R_v	the humidity intrusion rate.
T_{Dry}	the time that is needed to let the humidity from within the wall to dissipate, i.e. the time when the wall is dry to apply the stucco.
T_{Dry}^{rule}	the time that is needed to let the wall dry according to the regulations.
$T_{Dry}^{pred}(n)$	the time that is needed to let the wall dry according to the predictive model using n tests.
v_0	boundary condition for the humidity.
v_c	the critical humidity level.
V	the humidity.
x_i	location of the measurement sensor.
$Z(t)$	the limit state function as a function of time.

Bibliography

- [1] *International society for bayesian analysis (isba) website*, <http://www.bayesian.org>.
- [2] *References for thomas bayes*, <http://www-groups.dcs.st-and.ac.uk/history/References/Bayes.html>.
- [3] *Wikipedia, the free encyclopedia*, <http://en.wikipedia.org/wiki/Category:Statistics>.
- [4] O'Hagan A. and Forster J., *Bayesian inference*, 2nd ed., Kendall's Advanced Theory of Statistics, vol. 2B, Oxford University Press Inc., 2004.
- [5] Vrouwenvelder A.C.W.M. and Straalen IJ. van, *Beoordeling van de veiligheid door middel van experimenten*, TNO-Report B-93-0066, TNO Bouw, December 1993.
- [6] Ada Decision Systems, 2710 Sand Hill Road, Menlo Park, CA94025,USA, *Dpl advanced version user guide*, 1992.
- [7] Wooff D., *Bayes linear methods 2-an example with an introduction to [bd]*.
- [8] Wit M.S. de, *Duurzaamheid met behulp van inspecties en onderhoud - voorbeeldstudie 2*, TNO-Report 2002-CI-R2075, TNO Bouw, May 2002.
- [9] ———, *Ongevallenreconstructie met behulp van bayesiaanse statistiek*, TNO-Report 2003-CI-M1039/WTM, TNO Bouw, December 2003.
- [10] Wit M.S. de and Wiel W.D. van der, *Empirische betrouwbaarheid*, Tech. report, TNO Building and Constructions Research, December 2004.
- [11] HUGIN Expert, *The hugin manual*, HUGIN Expert, Denmark, 6.2 ed.
- [12] Jensen F.V., *Bayesian networks and decision graphs*, Statistics for Engineering and Information Science, Springer-Verlag, 2001.
- [13] Berger J.O., *Statistical decision theory and bayesian analysis*, Springer series in Statistics, Springer-Verlag, 1980.
- [14] Savage L.J., *The foundations of statistics*, Wiley, 1954/1972.
- [15] Goldstein M., *Bayes linear methods-adjusting beliefs-concepts and properties*, no. 1.
- [16] ———, *Bayes linear analysis*, Encyclopaedia of Statistical Sciences **Update 3** (1998).
- [17] Norstrøm J.G., *Value of information design of control software*, Delft University Press, 2000.

- [18] Ditlevsen O. and Vrouwenvelder A., *Objective' low informative priors for bayesian inference from totally censored gaussian data*, Structural Safety (1994), no. 16, 175–188.
- [19] Vrouwenvelder T Ravenshorst G., Linden M van der and Kuilen JW van de, *An economic method to determine the strength class of wood species*, HERON (2005).
- [20] Cooke R.M., *Experts in uncertainty*, Oxford University Press, 1991.
- [21] Cooke R.M. and Bedford T., *Probabilistic risk analysis: Foundations and methods*, Cambridge University Press, 2001.
- [22] Vrouwenvelder A.C.W.M. Wit M.S. de, *Duurzaamheid met behulp van inspecties en onderhoud - een voorbeeldstudie*, Tno-report, TNO Bouw, 2000.

Index

- timber case, 5
- additive model, 6
- analytical approach, 15
- Bayes decision rule, 46
- Bayes linear, 20
- Bayes Risk, 46
- Bayes' rule, 3
- Bayes' theorem (for events), 2
- Bayes, Thomas, 2
- Bayesian method, 9
- bi-linear loss, 53
- conjugate approach, 15
- conversion error, 62
- data, destructive, 5
- data, non-destructive, 6
- decision problem timber case, 48
- decision tree approach, 49
- design approach, 49
- direct numerical integration, 18
- expected utility, 40
- fixed size test plan, 47
- frequentist interpretation, 1
- gamma distribution, 16
- General Method for Testing, 59
- HUGIN, 18
- humidity case, 60
- humidity relation, 62
- improper priors, 12
- inverse-gamma distribution, 16
- law of total probability(LTP), 3
- likelihood, 10
- likelihood-function, 3
- measurement accuracy, 62
- multiplicative model, 6
- normal inverted gamma model, 16
- prior information, 10
- producers approach, 49
- quadratic loss, 51
- Savages decision theory(SDT), 39
- subjective probability, 1
- update, 10
- value of information, 47