NTNU
Norwegian University of
Science and Technology

Master's thesis

2019

Norunn Ahdell Wankel

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

Norunn Ahdell Wankel

Hierarchical Bayesian modeling of wind related transmission line failures

April 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Hierarchical Bayesian modeling of wind related transmission line failures

Norunn Ahdell Wankel

Master of Science in Physics and Mathematics
Submission date: April 2019
Supervisor: Håkon Tjelmeland

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis is the result of my work in the course TMA4900 Industrial Mathematics, Master's Thesis, at the Norwegian University of Science and Technology (NTNU). It completes my master's degree in Applied Physics and Mathematics with specialization within Industrial Mathematics, and more precisely statistics. The work on this thesis was started in November 2018.

After an interesting summer job at Statnett in the summer of 2018 I asked if it was in their interest to have me writing a thesis on a topic that they are working with. To my delight, they said yes, and for this I would like to thank Thomas Trötscher, the Head of the Data Science department at Statnett. It is his colleague Øystein Rognes Solheim that have provided me with relevant data on wind and transmission lines and who has been my contact person throughout this period to answer questions regarding the data or their existing model. For that I am grateful.

The final task description and supervising was done by Håkon Tjelmeland, my supervisor at NTNU through both my specialization project and now master's thesis. In addition to weekly meetings he has been extraordinary flexible and available whenever there has been a question or problem to discuss. He has also shared thoughts and advises when coding the MCMC algorithm. For all of this I am extremely grateful.

As this thesis required computational intensive methods I have run the computationally heavy parts on a computational server for math students at NTNU. For help with questions and concerns regarding the computational server and parallel coding Per Kristian Hove, the leader and coordinator of the Technical group at the Department of Mathematical Sciences, has always been very helpful. Due to a power outage at NTNU the aforementioned server for where our computations were running went down and all computations stopped. As this happened fairly close to the master's thesis deadline, we found it a natural endpoint for the MCMC runs. This really made me realize how dependent my thesis and I are upon availability of electricity.

Last but not least I would like to thank my fellow math students for making the hours at school rather joyful, and my flat mates for keeping my mind off school when needed. My years here in Trondheim would not have been the same without you.

Norunn Ahdell Wankel
Trondheim, Norway
April 9, 2019

Abstract

Electricity and its availability is important in today's society. There are many reasons why components in the power system fail, with weather being one of them. In this report we formulate two Bayesian hierarchical models for wind related failures on overhead transmission lines. These models are based on an already existing model in use by the Norwegian Transmission System Operator. The number of failures along a line segment is assumed Poisson distributed, with an intensity parameter that is a function of wind speed, segment length, a common parameter for all lines and two line-specific parameters. The parameters are assigned prior distributions and an expression for the corresponding posterior distribution is then obtained by combining the likelihood and prior. Our two models differ only in terms of the likelihood, where one model considers failures on segment-level, while the other only includes the total number of failures along a line. Since we are not able to work analytically with the posterior distributions MCMC methods are used.

An MCMC updating scheme consisting of Metropolis-Hastings and Gibbs updates are run for both models on simulated failure data. The posterior 90% credible interval is for almost all parameters narrower for the line-wise model than the segment-wise model. In both cases a positive correlation between the two line-specific parameters are seen, which indicates that one in the future should consider to include another type of prior for these parameters that allow for a correlation parameter. The generated parameter values from the MCMC runs are then used to predict the probability of at least one failure for future times and compared to the true probabilities of failures. For the comparison the logarithm of the absolute error of the predictions are calculated and plotted together for the line-wise and segment-wise model. However, there seem to be no distinct difference between them. This indicates that there might be nothing or little to gain for a Transmission System Operator in terms of better predictions if they improve the reporting on the exact location of failures.

Sammendrag

Elektrisitet er viktig i dagens samfunn og noe vi nærmest tar for gitt at alltid skal være tilgjengelig. Det kan være mange grunner til at deler av kraftsystemet feiler, og vær er en av dem. I denne masteroppgaven formulerer vi to bayesianske hierarkiske modeller for forbigående feil på høyspentlinjer som er forårsaket av vind. Disse modellene er basert på en allerede eksisterende modell som brukes av systemansvarlig i det norske kraftsystemet. Vi lar antall feil for et linjesegment for en gitt time være poissonfordelt. Den tilhørende intensiteten er en funksjon av vindhastighet, linjesegmentlengde, en felles parameter for alle linjer samt to linjespesifikke parametre. Disse parametrene gir vi en apriorifordeling og ved å kombinere likelihood og apriorifordelingen finner vi et uttrykk for den tilhørende aposteriorifordelingen. Modellene vi formulerer har ulik likelihood, hvorav den ene belager seg på feil på segmentnivå, mens den andre kun inkluderer totalt antall feil for en linje sett under ett. Fordi vi ikke klarer å jobbe analytisk med aposteriorifordelingene velger vi å ta i bruk Markov chain Monte Carlo-metoder for å simulere fra fordelingene.

Vi simulerer feildata og bruker dette som input til MCMC-kjøringer som består av parameteroppdateringer basert på Metropolis-Hastings og Gibbs algoritme. Det viser seg at linjemodellen har de smaleste aposteriori 90%-kredibilitetsintervallene for nesten alle parametrene i modellen. I begge tilfeller observer vi at det er en klar positiv korrelasjon mellom de to linjespesifikke parametrene. Dette indikerer at man i en eventuell fremtidig modell burde vurdere en annen apriorifordeling for disse parametrene som tar høyde for en korrelasjonsparameter. De simulerte parameterverdiene fra MCMC-kjøringene brukes for å estimere sannsynligheten for minst en feil for fremtidige timer og evalueres mot de sanne sannsynlighetene for feil. Vi sammenligner modellene ved å plotte logaritmen av absolutt feil for alle prediksjoner. Ut ifra dette kan vi ikke se noen tydelig forskjell mellom modellene. Dette er dermed en indikasjon på at det i prediksjonsøyemed er lite eller ingenting å hente for systemansvarlig i kraftsystemet ved å forbedre rapportering av feildata til å gi eksakt lokasjon for feil.

1 Introduction

Electricity is a vital part of the modern life, it is among others needed in households to charge our phones, light our homes, keep food and drinks cooled in the refrigerator, and for public services and businesses to operate machines and keep servers and computer systems running. However, the availability of electricity at all times is something we in developed countries often take for granted. A prerequisite for this is a well-functioning power system, and in particular a reliable electrical grid to transport the electricity from where it is generated and all the way to the end users (Ward, 2013).

In Norway the state-owned company Statnett SF has the role as the nation's Transmission System Operator (Norwegian water resources and energy directorate, 2016), meaning that they have several responsibilities regarding the power system. One of the most important ones is to continuously upheld the balance between consumption and generation of electricity. With this also the importance of a reliable system emerges. The electrical grid is divided into the transmission grid, the regional grid and the distribution grid, where the transmission grid is the one for long-distance transportation carrying high-voltages, i.e. 132kV-420kV. The rest of the grid, which is often connected to the central grid, is for power lines carrying lower voltages. These lines reach out to smaller consumers and industries, such as normal households and service industries (Norwegian Ministry of Petroleum and Energy, 2019a). It is not realistic to have no interruptions in the supply of electricity, as this would require one to invest too much if one considers the costs versus the benefits (Norwegian Ministry of Petroleum and Energy, 2019b). With a nation-wide grid of a total length of around 130 000 kilometres, where approximately 11 000 of these are part of the transmission grid operated by Statnett, it is only a question of when and where a component is going to fail next.

An electrical grid often consists of both overhead lines and underground cables, in addition to transformers, control centers and other equipment (Ward, 2013), and hence several components that have a propensity to fail. In this report we look at failures due to wind on overhead transmission lines, and only the ones classified as temporary failures. In this particular setting a failure is roughly the same as something is not functioning or it has a reduced ability to function as it is supposed to do, see Statnett's own failure statistics and definitions online for more details (Statnett SF, seksjon Feilanalyse, 2015). The failures are divided into temporary and lasting ones, in addition to be categorized based on the reason for failure such as weather, vegetation, birds and human errors. A temporary failure is defined as a failure which does not require any specific repair, maybe just some adjustments (Statnett SF, seksjon Feilanalyse, 2015). Even though there might be a wide variety of possible reasons for why components fail, most of the temporary failures on overhead transmission lines are known to happen due to or when we experience severe weather (Solheim et al., 2016). In particular, wind, lightning and ice are the three main weather factors. To be able to find a way to use the weather report to predict the weather related failure probability for each line would be of great help in the reliability work of system operators.

Until now, Statnett has indeed implemented models for predicting the probability of temporary failure on overhead transmission lines due to wind and lightning, in two separate models. These are presented in detail in one article each, see Solheim et al.

(2016) regarding the wind model and Solheim and Trötscher (2018) for the lightning model. Based on these articles we summarize the structure of the models, as it is the starting point for developing our own models later:

By sectioning each power line into segments, defined by the part of the line between two consecutively power towers, they treat a line as a series system. Assuming all segments independent, and having available estimated weather data on an hourly basis they connect one or more weather parameters to the probability of failure of a line segment within an hour. This is based on the concept of fragility curves, and is in this setting a function which maps the weather in terms of wind or lightning exposure to the probability for a line to fail. The cumulative lognormal function is the one used. They treat the parameters of this curve as constant for all times, and hence one ends up with only one set of parameters for each power line. To be able to determine the values of these parameters they have chosen to do this by calibrating the hourly probability of failure for a line to suit actual observed failure rates.

For the calibration one formulates a simple Bayesian model for each transmission line, assuming that the number of annual failures are independent and identically Poisson distributed. The corresponding intensity parameter is given an exponential prior distribution with an intensity calculated based on information of overall failures per year per 100 km, and adjusted to each line based on the length of the line, the type of failure and type of weather. This results in a posterior gamma distribution. Based on observed failure data the posterior mean is easily calculated, and this value is used for calibration. Optimization is done by finding the parameters of the fragility curves such that the overall posterior yearly failure rate is "close" to the expected number of annual failures based on the hourly probability time series. In addition, a term for minimizing the Brier score (Brier, 1950) can be added so that complete wrong predictions are penalized a lot. Having found each lines parameters, prediction is done by using weather forecast on an hourly basis as input to the fragility curve. One then gets the predicted probability of failure for a line for the same amount of hours one has weather forecasts for.

The models which Statnett use today have some constraints. Among others they are not capturing the uncertainty in the parameters, and hence also in the predictions, well. This is due to the fact that they are only optimizing with respect to the posterior mean annual failure rate, when in fact there is an uncertainty in this parameter, and hence one then do not get a feeling of how much the estimates of the parameters in each fragility curve could in fact vary.

In this report we propose another type of model, namely a Bayesian hierarchical model, yielding instead of just estimates for the parameters in the cumulative lognormal rather the posterior distribution of them. Naturally this also gives information about the uncertainty in each parameter. We only consider temporary failures due to wind and use Statnett's wind model as a natural starting point. In addition we make some minor changes to the definition of wind exposure used in their model to suit our model. Also, while in their model the line specific parameters would just be estimates based on information from that specific line, in our model we let all data influence these parameter values. This can be a huge advantage if for instance one does not have that much data for all lines.

Overall the lines mostly do not fail. This is of course desirable with respect to reliability, but this also creates a problem when creating a model. If we run the model on little data, the total number of failures is small. However, to avoid running with true observed failures for many years, which would have been computationally demanding for this project, we rather look at six months and simulate data for this period. This is done such that we get approximately the same total number of failures in this shorter time period as for the whole true data set.

The main objective of this thesis is to construct Bayesian hierarchical models for the probability of failure along power lines. We formulate two similar models, one for which we only consider failures reported for each line and one where we assume more detailed failure data, and hence failures on segment level. To be able to get any information about the parameters in the models we use Markov chain Monte Carlo (MCMC) methods to sample from the corresponding posterior distributions. The uncertainty of the parameters are assessed through credible intervals, and seen in light of the simulated data used. In particular, we are interested in how our models can be used to predict the probability of failure for future hours. We predict the probability of at least one failure for some future times based on the MCMC runs. Lastly, we compare the predictions made and comment on how one could score predictions in practice.

This report is structured as follows: In Section 2 we introduce the theory and concepts used for our model. The focus is mainly on Bayesian hierarchical modeling and MCMC methods. In section 3 we have a closer look at the data provided by Statnett, which is wind data and failure data for several power lines in Norway, along with information of the length of each line segment. Then we present our models in detail in Section 4, which we end by introducing the simulated failure data actually used for the MCMC. In Section 5 we state the algorithm and specific choices made for generating samples from the corresponding posterior distributions. Following this, implementation notes are found in Section 6. Some results from runs of the implemented code using simulated failure data are shown and discussed in Section 7. We end this report with some closing remarks and thoughts about how our models and algorithms can be of further use.

2 Theory

In this section we present the concepts and theory needed for understanding the idea behind our Bayesian hierarchical models which we later formulate in Section 4, in addition to methods used to be able to use such models in practice. Since these type of models are based on the Bayesian statistical view, we start by giving a brief introduction to and motivation for Bayesian statistics, and thereby hierarchical models in particular. For such models we tend to get complicated distributions which we need to sample from, and so MCMC methods are often used. The basics of Markov chain Monte Carlo are presented and the two most known algorithms, Metropolis-Hastings and Gibbs, are further given in more detail. An overview of how one can use the output from an MCMC run to assess whether the sample generated is indeed from the desired distribution and how to sample efficiently is also given. In addition, we briefly introduce how predictions can be made based on the generated parameter values from an MCMC run. We wrap up this section by presenting some probability distributions with parameterizations corresponding to the ones we use in our model.

2.1 Bayesian hierarchical modeling

If we are interested in understanding and describing a phenomenon, an experiment or a problem or making future predictions we are in need of a model. In statistics the model is defined through probability distributions and assumptions, for instance regarding independencies of random variables. Already in the choice of probability distribution we have assumed that the data can be described by such a distribution. Note that most models are simplifications of a (real-world) problem, and are in most cases wrong, since the true underlying process is unknown. However, it still makes sense to create a model. For instance, we might assume that a Poisson distribution fits the data if we are dealing with counting data, or an exponential distribution for lifetimes. However, how we treat the parameters in a statistical model is dependent upon which approach we use, the frequentist or the Bayesian one.

In the frequentistic way, the parameters are treated as fixed, but unknown (Bolstad, 2007). We are then interested in obtaining estimates, for instance maximum likelihood estimates, of the corresponding parameters. In addition we often associate a (1-α)% confidence interval to each estimate. Recall that since the parameters are fixed, a parameter either will be in the interval or not, and hence we can not talk about a for instance 95% probability of the parameter being in the corresponding confidence interval. However, if repeatedly obtaining samples from the same distribution and creating a (1-α)% confidence interval each time, the interval is going to cover the true parameter value (1-α)% of the times (Walpole et al., 2012). This summarizes the way one explains the term "probability" in the frequentist approach, namely as the long-run relative frequency when performing the sampling infinitely many times.

Bayesian statisticians on the other hand treat the parameters in the model as random, and hence unknown (Bolstad, 2007). This also requires us to state some sort of belief about the parameters, known as the prior distribution. The knowledge one has about these parameters might be influenced by how much experience one has with the domain, if one has knowledge of any similar problems or if one just has a strong personal belief. The prior distribution is also called subjective since it differs from person to person and time to time, since the information available for different persons and at different times naturally might differ. The prior distribution might be rather flat to indicate no or little knowledge or rather specified if one has a stronger belief. We are then not only interested in estimates, but also in the marginal posterior distributions of each parameter, hence the distribution of the parameter conditioned on observed data (Gelman et al., 2014). To assess uncertainty of a parameter we can look at credible intervals, which are intervals where we in contrast to frequentistic confidence intervals can actually state that the probability of a parameter to be within it is for instance 95%.

In many ways, Bayesian statistics is well incorporated in daily life, because we often like to refer to events as having a certain probability of happening. In addition we often have some knowledge of similar problems from before, which is easily incorporated in the Bayesian setting in terms of the prior distribution. Note however that even though frequentists do not assume a prior distribution, we always makes some assumptions or to some degree subjective choices when creating a model anyway, which among others include decisions of what data to include and which type of probability distributions we choose, as pointed out in (Gelman et al., 2014). The rest of the theory section is devoted to the Bayesian approach, as this is the basis for the models we formulate later on in this report.

In Bayesian theory the Bayes' rule is of course central. Let x denote the data, which we believe to be generated from a sampling distribution with parameter θ, denoted $f(x|\theta)$. Our prior belief of this parameter is contained in the prior distribution, which we denote $f(\theta)$. Having observed data, $f(x|\theta)$ can be interpreted as a function of the parameter θ, and is then called the likelihood function. We let $f(x,\theta)$ denote the joint distribution of the parameter and data. To update our belief about the parameter, both the likelihood and prior is combined through what is known as Bayes' rule in the following way (Gelman et al., 2014)

$$f(\theta|x) = \frac{f(x,\theta)}{f(x)} = \frac{f(x|\theta)f(\theta)}{f(x)}, \tag{2.1}$$

to obtain the posterior distribution $f(\theta|x)$. However, we often only need to consider the unnormalized version of the posterior distribution. The right hand side of (2.1) simplifies to

$$f(\theta|x) \propto f(x,\theta) = f(x|\theta)f(\theta), \tag{2.2}$$

which is proportional to $f(\theta|x)$ in terms of θ. Note that the term $f(x)$ is only considered a constant after the data x is observed. Keep in mind that both x and θ can denote a vector of random variables, hence that they might be multivariate variables.

To fully specify a model one only needs to specify the joint distribution of all random variables, since as seen before from (2.2) this joint distribution is proportional to the posterior distribution. However, to specify the joint distribution in just one step is not straightforward as this would require us to choose a multivariate distribution that captures how each of the variables are correlated. For the simplest Bayesian model we can rather specify the model in terms of the likelihood $f(x|\theta)$ and the prior $f(\theta)$. If letting x consist of N variables $x_1, ..., x_N$ it is useful to assume conditional independence such that the likelihood can be written as a product, hence as $f(x_1, ..., x_N|\theta) = \prod_{i=1}^{N} f(x_i|\theta)$. To go from a standard Bayesian model to a hierarchical one, the parameters of the prior distribution, which we denote by ϕ, is treated as unknown and are assigned a distribution (Ntzoufras, 2009). This distribution is referred to as the hyperprior and we denote it $f(\phi)$. It is useful to continue the hierarchical structure on the joint prior distribution, such that $f(\theta,\phi) = f(\theta|\phi)f(\phi)$. If assigning a prior on the parameter γ of the hyperprior distribution, we get another level of hierarchy, the so called hyper-hyperprior (Lee, 2012). Let this be denoted $f(\gamma)$, and then with the conditional distribution $f(\phi|\gamma)$ on the hyperprior level. See Figure 1 to see an overall illustration for the three types of models mentioned above.

For the case of a 2-stage hierarchical prior $f(\theta,\phi)$ the posterior distribution is given as (Kroese and Chan, 2014)

$$f(\theta,\phi|x) \propto f(x|\theta)f(\theta|\phi)f(\phi).$$

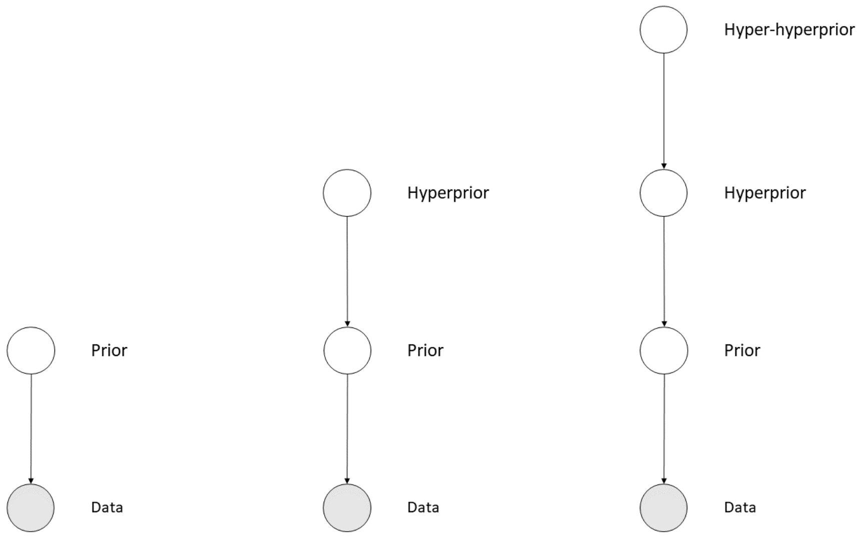

Figure 1: Simple illustration of three Bayesian models. From left to right we have a standard Bayesian model, a 2-stage Bayesian hierarchical model and a 3-stage Bayesian hierarchical model. The grey nodes indicates the data level of the model, hence the observable variables. The rest of the nodes represent one or more parameters. One must assign probability distributions to all random variables to fully specify the corresponding model.

In the same manner for the 3-stage prior we get

$$f(\theta, \phi, \gamma|x) \propto f(x|\theta)f(\theta|\phi)f(\phi|\gamma)f(\gamma).$$

To find the marginal posterior distribution for a parameter, this is simply the same as integrating out all other variables from the posterior. So for the 2-stage prior model

$$f(\phi|x) = \int f(\theta, \phi|x)d\theta.$$

would for instance be the marginal posterior distribution for the variable ϕ.

Even though we can make complex models, they are often easily represented through directed acyclic graphs, so-called DAGs. A DAG consists of nodes with directed arrows between them, in such a way that there is no cycle (Kroese and Chan, 2014). Here the layers are commonly separated, such that all units or parameters for one level are located on the same horizontal line. The nodes represent different stochastic variables, which might be parameters or observable variables. An example of the standard Bayesian model is depicted in Figure 2, in addition to one for the 2- and 3-stage hierarchical models in Figures 2 and 3, respectively. The nodes for the data level is colored grey, which are the ones we observe.

Often the data exhibits a natural hierarchical structure, as Hoff (2009) mentions is the case if we for instance are looking at patients in several different hospitals or at persons within counties which again lies within a region, and where the region again belongs to a country. Hence, when having data at an individual level that are somewhat grouped on a higher level. Let us look at an example to motivate the use of the different model types. If one is formulating a model for a school test result, the simplest one would be to only consider one school. This corresponds to Figure 2, with θ being a parameter specific for this particular problem, hence this particular school. However if rather looking at J schools in total, with N_j observations from school j, we have the scenario depicted in Figure 3. Here θ_j denotes the school-specific parameter for school j, but there is also a hyperprior-level since we assume that these school-specific parameters come from a common population with parameter ϕ. In other words, we believe the school parameters to be somehow "alike", maybe in the sense that they are all located in the same county. If one had additional data with test results from schools from several counties, one could extend the model as shown in Figure 4. Here we have I counties and J schools. We let all school-specific parameters be dependent on a county-specific parameter ϕ_i. Again we assume there is some connection between all counties, and hence all schools, by having a hyper-hyperprior level, with γ as the overall parameter. This could for instance have been the overall mean. Note that there is no need to add an extra layer in the hierarchical structure if one has no extra information that makes this desirable. For instance if only looking at test results from one school, one does not need both a school-specific mean and overall mean. Then the overall mean would just end up being equal to the school-specific one since only having data from one school.

The real strength of Bayesian hierarchical models is that we use all data to get information about all unknown parameters. This stands in contrast to the classical way of using data for only one group for estimating variables related to that group. The latter approach is for instance very sensitive to small data sets, which then will give a large

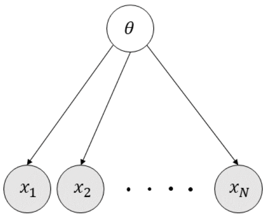

Figure 2: An example of a standard Bayesian model consisting of a data level and prior level. The joint distribution of data and parameters then consists of the likelihood corresponding to the distribution for the data level in addition to a prior distribution for the parameters of the sampling distribution.

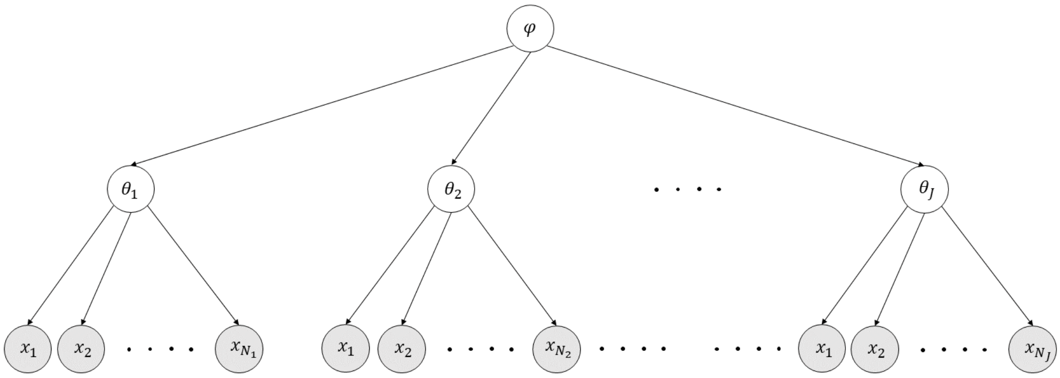

Figure 3: An example of a Bayesian model with a 2-stage hierarchical prior. It consists of a data level, prior level and hyperprior level. Here there are J groups, and hence J group-specific parameters θ_1,...,θ_J. In addition, ϕ is a common parameter for all groups.

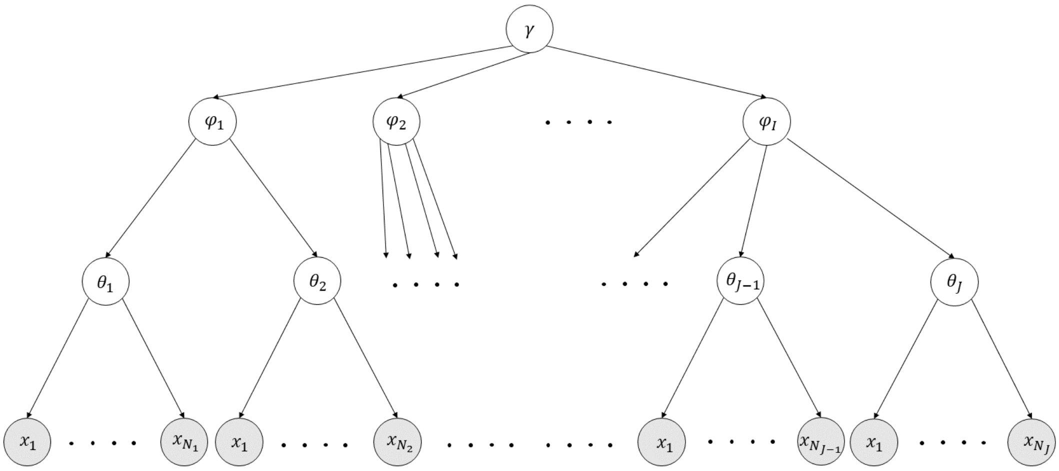

Figure 4: An example of a Bayesian model with a 3-stage hierarchical prior. The model consists of a data layer, prior level, hyperprior level and hyper-hyperprior level. Here there are two levels of groups, first J groups with a group-specific parameter θ_j, and then these parameters are again part of a larger group i with parameter ϕ_i. Let γ denote the overall common parameter.

uncertainty in group-specific estimates. In Bayesian models however, we incorporate information about all groups at once, such that even though we might have little data for one group, we use all the data to get values for the common parameters, and this again affects the prior parameter. There is a tendency of group-specific parameters being shrunk towards the corresponding population mean, how much is however dependent on the variance between all group parameters in that population. In addition, the less data we have for one group, the more the corresponding parameter is going to be shrunk (Ntzoufras, 2009). What might been seen as an additional strength is the use of prior distributions, which enable us to use all information available and to somewhat restrict the plausible values of parameters based on this knowledge. If having informative priors, this also reduce the posterior uncertainty about the parameters.

Prior information can be chosen as to reflect all information available or rather the lack of knowledge. For informative priors it is common to choose a prior that makes calculations and computations convenient, i.e. as a distribution within a well known parametric class. As mentioned in one of the examples in Hoff (2009), the prior information we have about a parameter might only be that we have some sort of interval we believe it to have a large probability of being within, and maybe we have a feeling of its expected value. Based on such info, we can however construct several priors that captures this information. One then often chooses one that makes computations convenient, for instance by choosing a conjugate prior. For this particular choice the prior and posterior end up having the same parametric form, see Gelman et al. (2014) for a more precise definition of conjugate priors.

Sometimes we rather want the data to dominate the analysis and therefore let the prior distribution be as uninformative as possible. This might be the case if one has no particular prior knowledge or if one is doing research and one wants to test a hypothesis where one would like the result not to be much influenced of one's own prior belief (Gelman et al., 2014). Naturally, uninformative priors can be described as flat, which is indeed the case for a common uninformative prior, namely the uniform distribution on an infinitely wide interval. Note that this is what one classifies as an improper prior, meaning that it is does not integrate to 1, and hence is not what we would describe as a proper prior. For variance parameters however, a common uninformative prior is $f(v) \propto \frac{1}{v}$, where we let v denote a variance parameter. It is important to be aware of that using improper priors does not necessarily give proper posterior distributions. One should therefore check that the corresponding posterior distribution is indeed proper, ideally by assuring that the integral $\int f(\theta|y)d\theta$ is finite. Inference based on the posterior distribution only makes sense if it is indeed a proper probability distribution.

There are also many other choices for priors, among others Jeffrey's prior and weakly informative ones. Jeffrey's prior is constructed in such a way that it is invariant to different parameterizations, with the intention that the belief expressed through the prior should be the same no matter how we choose to parameterize. A weakly informative prior only contains some information, and hence less than what might actual be the information available, and is therefore called "weak" (Gelman et al., 2014). It is a proper prior such that it in fact restricts the range of plausible values for a parameter.

After having observed data x we often aim to use these models to predict future, and yet unobserved, observations. For the Bayesian approach what we end up with is not merely an estimate but rather the distribution of the future observation. This is called the posterior predictive distribution. We let the this future observation be denoted $\tilde{x}$ and we assume that $\tilde{x}$ and the observed data x is independent given the parameter θ. By integrating over all possible parameter values θ the posterior predictive distribution is obtained as (Lesaffre and Lawson, 2012)

$$\begin{aligned} f(\tilde{x}|x) &= \int f(\tilde{x}, \theta|x) d\theta \\ &= \int f(\tilde{x}|\theta, x) f(\theta|x) d\theta \\ &= \int f(\tilde{x}|\theta) f(\theta|x) d\theta. \end{aligned} \tag{2.3}$$

It is clear that this distribution takes the uncertainty of θ into account, and to get some measure of the uncertainty in the prediction of $\tilde{x}$ one often looks at posterior predictive intervals. These are no more than credible intervals for $\tilde{x}$, with a $(1-\alpha)\%$ probability of the future observations to be within the interval. Naturally, the wider the interval, the more uncertain we are about which value a new observation will have.

If one is observing data at different times, say for instance that one has collected one set of data x_1 and at a later time one collects one more x_2, the model can be updated in-between the collections as to account for all available information. This is done by using the posterior distribution $f(\theta|x_1)$ as a prior for θ before collecting the second data set. To see that this is a natural choice consider the posterior distribution, with conditioning on both data sets. This yields

$$\begin{aligned} f(\theta|x_1, x_2) &\propto f(\theta, x_1, x_2) \\ &= f(x_2|\theta, x_1) f(\theta|x_1) \\ &= f(x_2|\theta) f(\theta|x_1) \end{aligned}$$

where we again have used the assumption of conditional independence between observations, and hence also between data sets. Here $f(x_2|\theta)$ can be seen as the likelihood-term and $f(\theta|x_1)$ as the prior term. The updated version of the posterior distribution is then in general easily obtained by combining the newest posterior distribution together with the likelihood for the last data set. For a total of D data sets this yields the following updating scheme (Ntzoufras, 2009)

$$f(\theta|x_1, ..., x_D) \propto f(x_D|\theta) f(\theta|x_1, ..., x_{D-1}) \propto f(\theta) \prod_{i=1}^{D} f(x_i|\theta). \tag{2.4}$$

2.2 MCMC

As mentioned earlier when creating a Bayesian model we are interested in the posterior distribution since we for instance can use this to predict values for future observations.

However, even though we might choose standard or relative simple distributions for each unit on all levels in the model, the product of conditional distributions that makes up the posterior might yield a non-standard and complex distribution. An iterative sampling method based on Markov chains, named Markov chain Monte Carlo (MCMC), has become popular from the 1990s due to its flexibility and because of the increasingly availability of computing power (Ntzoufras, 2009). The idea of this method is to construct a Markov chain, which is both irreducible and aperiodic, and with a limiting distribution f equal to the target distribution (Givens and Hoeting, 2013). By target distribution we refer to the distribution we would like to obtain samples from. In the Bayesian setting this is most often the posterior distribution. There are several MCMC methods, but we focus on the first developed method known as the Metropolis-Hastings (MH) algorithm and a special case of it, namely the Gibbs sampling algorithm (Chib and Greenberg, 1996).

2.2.1 Metropolis-Hastings algorithm

Recall that a Markov process is a stochastic process which has the Markov property. We denote the process by $\{\theta^{(m)}, m \in I\}$, where I is called the index set, and where S is the state space, hence a set containing all possible values of $\theta^{(m)}$ (Rubinstein and Kroese, 2008). We consider only a discrete index set and the process is then named a Markov chain. The Markov property states that

$$f(\theta^{(m+1)}|\theta^{(m)}, \theta^{(m-1)}, ..., \theta^{(1)}, \theta^{(0)}) = f(\theta^{(m+1)}|\theta^{(m)})$$

which means that the distribution of $\theta^{(m+1)}$ is independent of all past states, given the current one $\theta^{(m)}$. The random variables in this chain is in our setting typically d-dimensional vectors consisting of all parameters of a Bayesian model. Hence, we have $\theta^{(m)} \in \mathbb{R}^d$. The index set is in terms of MCMC methods equal to the non-negative integers such that $I = \{0, 1, 2, ...\}$. We let m denote the iteration number. The state space can be either discrete or continuous, and we continue with the continuous state space only. However, the discrete case is very similar.

Let $p(\theta^*|\theta)$ denote the transition kernel of a Markov chain, which is in fact a mixed distribution, and which is governing the probability of moving from the current state θ and into certain regions of the state space. Note that in the continuous case we can generally not speak of the probability of moving from one point to another, as is the case if having a discrete state space. For f to be a stationary distribution of a Markov chain with corresponding transition kernel p it must satisfy (Gamerman and Lopes, 2006)

$$f(\tilde{\theta}) = \int f(\theta)p(\tilde{\theta}|\theta)d\theta. \tag{2.5}$$

One can easily verify that if p satisfies what is known as the detailed balance conditions, which is the same as the chain being reversible, then f also necessarily satisfies (2.5). The detailed balance conditions are given by

$$f(\theta)p(\tilde{\theta}|\theta) = f(\tilde{\theta})p(\theta|\tilde{\theta}), \text{ for all } \theta, \tilde{\theta} \in S. \tag{2.6}$$

To see why this holds one can simply integrate (2.6) with respect to θ (Gamerman and Lopes, 2006), and by noting that $\int p(\theta|\tilde{\theta})d\theta = 1$ this yields the same as equation (2.5) for stationarity.

The Metropolis-Hastings algorithm is constructed in such a way that the corresponding transition kernel of the Markov chain indeed satisfies the detailed balance conditions, and hence has then automatically the required stationary distribution. The transition kernel is in the Metropolis-Hastings setting given as

$$p(\tilde{\theta}|\theta) = \begin{cases} q(\tilde{\theta}|\theta)A(\tilde{\theta}|\theta) & \text{for } \tilde{\theta} \neq \theta, \\ 1 - \int q(\check{\theta}|\theta)A(\check{\theta}|\theta)d\check{\theta} & \text{for } \tilde{\theta} = \theta, \end{cases} \tag{2.7}$$

where q denotes a proposal density and $A \in [0, 1]$ an acceptance probability. This corresponds to the procedure of generating a proposal for the next state from the proposal distribution q, and to accept this proposal as the next state with a probability equal to A. However, if not accepted, the next state is set equal to the current one, hence we get $\tilde{\theta} = \theta$. In the latter case we can actually speak of the probability of ending up in the same state since this probability is clearly non-zero, and it is equal to

$$1 - P(\text{proposing another state and accepting the proposal}) = 1 - \int q(\check{\theta}|\theta)A(\check{\theta}|\theta)d\check{\theta}.$$

By letting the transition kernel be on the form given by (2.7) we need an expression for the acceptance probability. This is chosen such that the transition kernel fulfills the detailed balance conditions. There are in fact several choices, but the one chosen for Metropolis-Hastings due to optimality reasons (Gelman et al., 2014) is

$$A(\tilde{\theta}|\theta) = \min\left[\frac{f(\tilde{\theta})q(\theta|\tilde{\theta})}{f(\theta)q(\tilde{\theta}|\theta)}, 1\right] \tag{2.8}$$

where we assume that the chain is in a valid state at all times, i.e. such that $f(\theta) > 0$. See Chib and Greenberg (1995) for more details. However, to ensure convergence towards the stationary distribution f some conditions must be met, namely irreducibility and aperiodicity (Chib and Greenberg, 1995). Since the requirements of irreducibility and aperiodicity are most often met, we do not linger more on this. One should however give it a thought when applying MCMC as to if one has the ability to explore the state space.

For a target distribution f, which is most often the posterior distribution in a Bayesian setting, we choose a proposal density q, and generate values of the chain as described above with the corresponding acceptance probability given in (2.8). For the case when our target distribution is the posterior distribution $f(\theta|x)$ the acceptance probability is given as

$$A(\tilde{\theta}|\theta) = \min\left[\frac{f(\tilde{\theta}|x)q(\theta|\tilde{\theta})}{f(\theta|x)q(\tilde{\theta}|\theta)}, 1\right]. \tag{2.9}$$

Note that in the construction of the Markov chain above, the target distribution is indeed assumed to be a proper probability distribution. As mentioned before, if using improper priors in the model, we are not guaranteed a proper posterior. If the posterior is indeed improper the above is not applicable. We do not go any deeper into the theoretical foundation of MCMC here as this would have been overkill with respect to the aim of this thesis, which is MCMC in an applied setting. For the very interested reader, (Tierney, 1994) and (Chib and Greenberg, 1996) give at least a somewhat thorougher explanation, see (Chib and Greenberg, 1995) for details on the MH algorithm in particular.

To summarize the Metropolis-Hastings algorithm we outline the algorithm in pseudocode below. We let here our target distribution be the posterior distribution.

1. Set an initial value $\theta^{(0)}$ for the Markov chain.
2. For m=1,2,....
    (a) Propose a potential new value for iteration number m by drawing a value θ^* from the proposal density $q(\theta^*|\theta^{(m-1)})$.
    (b) Calculate the acceptance probability $A(\theta^*|\theta^{(m-1)})$ given by (2.9).
    (c) Generate a value u from the Unif$[0,1]$. If $u < A(\theta^*|\theta)$ accept the proposal and set $\theta^{(m)} = \theta^*$, else reject the proposal and set $\theta^{(m)} = \theta^{(m-1)}$.

Note that in the expression for the acceptance probability A in (2.8) the target distribution only appears as part of a fraction, and hence any normalizing constants cancel out. Thus it is only necessary to know the unnormalized form of the target distribution, which is indeed the case for Bayesian models where the posterior often has a computational intractable normalizing constant.

In the above discussion we described the algorithm for the case of updating θ in one step, meaning that if θ is a d-dimensional parameter vector, we need a d-dimensional proposal density. However, it is also possible to update one element of the vector at a time, or several together in blocks. The target distribution is in all cases the posterior $f(\theta|x)$. We come back to this possibility later in the discussion of hybrid algorithms.

2.2.2 Gibbs sampler

A special case of the Metropolis-Hastings algorithm is the Gibbs sampler. Let again $\theta = (\theta_1, ..., \theta_d) \in \mathbb{R}^d$, but now all elements of θ are updated sequentially at each iteration. Each element is sampled from a univariate conditional distribution, given the rest of the elements. When sampling from the posterior distribution each univariate distribution is then equal to the full conditional $f(\theta_i|\theta_{-i}, x)$. Here θ_{-i} denotes all parameters except for element θ_i. The algorithm is then as follows (Givens and Hoeting, 2013):

1. Set initial value $\theta^{(0)} = (\theta_1^{(0)}, ..., \theta_d^{(0)})$ for the Markov chain.
2. For m=1,2,...

    Let θ be the most updated value, hence to begin with we have $\theta = \theta^{(m-1)}$.

    For $i = 1, ..., d$

(a) Sample θ_i^* from $f(\theta_i^*|\theta_{-i}, x)$.

(b) Set $\theta_i = \theta_i^*$, such that $\theta = (\theta_1, ..., \theta_{i-1}, \theta_i^*, \theta_{i+1}, ..., \theta_d)$.

We then have updated all components and get $\theta^{(m)} = \theta$.

Each update is in fact a Metropolis-Hastings update, hence we have in total d such updates per iteration. To see that this is the case let q_i be the proposal density for the update of element i. Right before updating the corresponding parameter θ_i the most updated parameter vector is then $\theta = (\theta_1^{(m)}, ..., \theta_{i-1}^{(m)}, \theta_i^{(m-1)}, \theta_{i+1}^{(m-1)}, ..., \theta_d^{(m-1)})$. Let θ_{-i} denote the vector of all the most recent updates of all elements expect number i. The proposal densities are in case of Gibbs equal to (Givens and Hoeting, 2013)

$$q_i(\theta^*|\theta) = \begin{cases} f(\theta_i^*|\theta_{-i}, x) & \text{for } \theta_{-i}^* = \theta_{-i}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

where the only element updated when proposing from q_i is the element θ_i. Note that the proposal is conditioned on all the most updated values of the other elements. The acceptance probability with this type of proposal yields $A = 1$, and hence all proposed values are accepted. This is verified by inserting the expression for proposal density (2.10) in the the expression for the acceptance probability (2.9), and by noting that $\theta_{-i}^* = \theta_{-i}$. This type of proposal distribution only makes sense if it is easy to sample from the full conditionals.

Note that the updates do not necessarily need to be for one parameter at a time. One can organize the elements in blocks, where each block can be either univariate or combining two or more parameters at an update stage. If only updating one parameter at a time we get moves only along the axes, while if having multivariate proposals we can move in other directions as well. This can be an advantage if several components are highly correlated (Gamerman and Lopes, 2006).

2.2.3 Hybrid algorithms

As already mentioned, in the MH algorithm one does not need to treat all parameters in one block, and thus having a multivariate proposal density. Analogously one does not need to update only one component at a time in the Gibbs sampler. Let $J \leq d$ be the number of blocks and θ_j the parameters in block j, where $j = 1, ..., J$. Each block has its own proposal density q_j, and hence different transition kernels. There are many ways in which order one can perform the updates. Either one can update one block at each iteration, or one can update all in a predefined order or one can update all in a random order. We only look at the one where we update in a predefined order. If updating all blocks once for each iteration, the corresponding algorithm is

```
1. Set an initial value $\theta^{(0)}$ for the Markov chain.
2. For $m$=1,2,....
   For $j$=1,2...,J
      Let $\theta$ be the most updated parameter vector at all times.
```

(a) Propose a potential new value for block j by drawing θ_j^* from the proposal density $q_j(\theta_j^*|\theta_j)$.

(b) Calculate the acceptance probability $A(\theta_j^*|\theta_j)$ given by (2.9).

(c) Generate a value u from Unif[0, 1]. If $u < A(\theta_j^*|\theta_j)$ accept the proposal and set $\theta_j = \theta^*$, else reject the proposal and let θ_j stay unchanged.

All the updated components now makes up $\theta^{(m)}$.

For a parameter that we can find a way to sample from its full conditional, a Gibbs step is a natural choice. With the algorithm outlined above one can combine Gibbs updates of one or several parameters with general MH updates of the rest of the parameters. Even with a cycle of different update types as above, the theory of MCMC can be extended to cover these cases. See Gamerman and Lopes (2006) for more details in the case of univariate component-wise updates. Then the overall transition kernel is now the one required to be irreducible and aperiodic.

2.3 Proposal distributions

A major part of the Metropolis-Hastings algorithm is the proposal distribution. There are several different choices for these distributions. If we let the proposal distribution for an update be equal the corresponding full conditional distribution, we get a Gibbs update as discussed above. However, it might not be easy to sample from such a conditional distribution, and one must then find other types of proposal distributions. We present here two possible choices for the proposal of a univariate variable.

2.3.1 Random walk

For a random walk the proposal is given as $\theta + \epsilon$, where θ denotes the current state and where ϵ is generated from some distribution $h(\epsilon)$ (Givens and Hoeting, 2013). The distribution h is often a normal, student-t or uniform distribution. The proposal density then satisfies $q(\theta^*|\theta) = h(\theta^* - \theta)$. If h is symmetric about zero, such that for instance $\epsilon \sim N(0, s^2)$, then we have symmetry, meaning that (Gamerman and Lopes, 2006)

$$q(\theta^*|\theta) = q(\theta|\theta^*).$$

Note that in this case, and any case where the proposal distribution is symmetric, the acceptance probability is equal to

$$A(\theta^*|\theta) = \min\left[\frac{f(\theta^*|x)}{f(\theta|x)}, 1\right] \tag{2.11}$$

as the proposal distribution factors in (2.9) cancel out.

We refer to a normal random walk as the case when h is a normal distribution with mean zero. This corresponds to proposing values from a normal distribution centered around the current value, and with a predefined standard deviation s. Such a parameter is called a tuning parameter (Ntzoufras, 2009). We can tune the proposal distribution by altering the value of s to make the algorithm more efficient. If having a large value for the standard deviation we allow for proposals possible far away from the current value. This might lead to many rejections of proposed values because of corresponding low posterior probabilities. However, if the standard deviation is very small we always propose values near the current value, and it will therefore take longer time to explore the whole parameter space. This results often in bad mixing. For a multivariate proposal yielding a random walk, the multivariate normal is an alternative (Ntzoufras, 2009).

2.3.2 Uniform proposal

The normal distribution has the whole real line as support and might be a good choice for parameters with the same support. However, let us now consider an alternative proposal distribution for positive-valued parameters. Instead of adding randomness to the current value we rather find a multiplicative approach. Let $r > 1$ be a fixed value and the proposal distribution a continuous uniform distribution on the interval $[\theta/r, r \cdot \theta]$. This is equivalent to letting a proposal be $\theta^* = \theta \cdot u_r$, where $u_r \sim \text{Unif}[1/r, r]$. The corresponding proposal density is then given as

$$q(\theta^*|\theta) = \frac{1}{\left(r \cdot \theta - \frac{\theta}{r}\right)} \text{ for } \frac{\theta}{r} \leq \theta^* \leq r \cdot \theta. \tag{2.12}$$

The value of r is typically set close to one. If $r = 1.1$ then it is approximate equal probability to propose a value that is smaller or larger than the current one. The corresponding acceptance probability is then found by inserting the proposal distribution (2.12) in the expression for acceptance probability in (2.9). This yields

$$A(\theta^*|\theta) = \min\left[\frac{f(\theta^*|x)}{f(\theta|x)} \cdot \frac{\theta}{\theta^*}, 1\right]. \tag{2.13}$$

An advantage of this specific type of proposal distribution is that if $p(\theta^*|\theta) > 0$ then we also have $p(\theta|\theta^*) > 0$, meaning that we can always come back to the current value in one move after having moved to a new value. One can easily think of cases when this is not the case, for instance for a proposal $\theta^* \sim \text{Unif}[(1-c)\theta, (1+c)\theta]$ for $0 < c < 1$. For this version one might get a proposal θ^* such that $q(\theta|\theta^*) = 0$, which means that the current value could not have been proposed from the state θ^*. As seen from the expression of the acceptance probability (2.9) this yields $A = 0$, which means that one rejects the proposal. Hence, this type of proposal can then potentially contribute to many unnecessary generations of proposed values. With the uniform proposal in (2.12) we ensure that there are no such excessive proposals made.

2.4 MCMC diagnostics and output analysis

Even if we are guaranteed convergence to the target distribution by the construction of the Metropolis-Hastings algorithm, given that some conditions are satisfied, this is only the case when $m \to \infty$. In practice we run the chain for a relatively long time, and treat the generated values as having approximately the target distribution (Chib and Greenberg, 1995). The convergence is not dependent on the chosen initial values, however the starting point might affect the first part of the iterations in such a way that they are unrepresentative (Kruschke, 2015). This is the case if starting at initial values far away from the mode of the posterior, as it takes time for the chain to move into a region of higher posterior densities. This period is called the *burn-in* period, and we usually discard the iterations that belong to this period.

As stated in Kruschke (2015) there are some goals when doing MCMC, and mainly that the samples we obtain are representative. For this to be the case the distribution of the chain must indeed be close to the target distribution, and such that all parts of the distribution have been explored. Another important aspect is that we need enough samples to obtain good estimates, i.e. estimates that do not vary much whether or not we had chosen other initial values. Last, but not least, we are in practice also concerned about the run time, which means that we want to get a sample in reasonable time and often using a reasonable amount of computational power. Hence, we would like to obtain a representative sample efficiently. To check representativeness both visual and numerical checks can be used. We look at some of them below.

2.4.1 Trace plot

Trace plot is a useful tool to visually check for convergence and to assess how many generated parameter values belong to the so-called burn-in period. A trace plot is a plot of iteration number along the x-axis with the corresponding generated values for a certain parameter along the y-axis (Ntzoufras, 2009). If the chain has converged we expect all values to lie within the same zone, and at least that there are no strong tendencies or periodicities. If for instance the graph is just moving in one direction it has at least not converged yet. However, one should be aware of that even though the trace plot might show sign of convergence, there is no guarantee that convergence is reached, as the chain might only have explored some part of the parameter space (Kruschke, 2015). We could for instance have a case with several modes, where one only explores one part of the parameter space corresponding to one of the modes if not run long enough.

To make it easier to determine if a chain has converged one can run several independent MCMC runs, starting from different starting points, and compare them. If the trace plots from all chains overlap each other after a certain number of iterations we have an indication of convergence (Kruschke, 2015). From this the burn-in period can be determined, and set such that all chains have converged after this period.

In the plots mentioned above we look at one parameter at a time. For the burn-in of the whole chain we are interested in an overall iteration number B for which all parameters seem to have converged. After all we are interested in the convergence towards the posterior distribution, which is the full joint distribution of all parameters given the data. One then has to choose this period based on all parameter trace plots to assure that they indicate convergence for all parameters after the burn-in period. We discard the samples from the burn-in phase, and use only the remaining values for inference.

From the trace plots we also can get a feeling of the mixing properties of our chain (Lesaffre and Lawson, 2012). If for instance there are long periods for which the chain hardly moves, it might have got stuck in one part of the parameter space. This can for instance happen if we always propose values very near the current one. Then it might take a lot of time to explore the rest of the parameter space. To improve mixing one should monitor the acceptance rates for the parameter updates.

2.4.2 Running mean plot

The running mean is the mean of the values generated for a parameter up to and including the current iteration (Lesaffre and Lawson, 2012). By plotting the iteration number versus the running mean corresponding to each iteration we see if the value stabilizes. If the chain has converged the running mean should eventually stabilize, and hence the running mean plot can be used in addition to trace plots to check if convergence seems reasonable. However, one then needs to generate values for far more iterations than is needed to reach convergence, as the mean otherwise will be greatly influenced by the unrepresentative values from the burn-in period.

2.4.3 Acceptance rate

To ensure good mixing using the MH algorithm one can keep track of the different parameter's acceptance rate, hence the proportion of all the proposed values that are accepted. Clearly, a low acceptance rate means that proposals often are rejected, and hence the chain stays at the same value for many iterations. Naturally the parameter space is then explored slowly, which also yields high autocorrelations (Ntzoufras, 2009). This is the case if choosing the tuning parameter in the proposal distribution to be too large. However, if the acceptance rate is very high then we accept almost all proposals and again this might yield slow mixing and highly correlated samples. In the normal random walk the tuning parameter is equal to the standard deviation s, and in the uniform proposal described earlier the corresponding tuning parameter is r. These parameters are often chosen arbitrarily at first, but by running the MCMC while monitoring the acceptance rates we can change the value of the tuning parameters before we start another MCMC run as to shift the acceptance rates in the desired direction. If one observes that the acceptance ratio is too high one increases the value of r or s, if too low one decreases the same tuning parameter values. Often acceptance rates in the region $20\% - 40\%$ (Ntzoufras, 2009) or $20\% - 50\%$ (Gamerman and Lopes, 2006) are recommended. These are however just guiding principles and not strictly necessary to obtain. For parameters updated using a Gibbs step the acceptance rate is necessary equal to 1 as all proposals are accepted, and we have no notion of tuning in that case.

2.4.4 Autocorrelation plot

An autocorrelation plot displays the correlation between different lags in the chain. The autocorrelation for lag k is then the correlation between generated parameter values that are k iterations apart, hence the correlation between $\theta^{(m)}$ and $\theta^{(m+k)}$ (Givens and Hoeting, 2013). If the chain is not mixing well this can be seen as a slow decay in the autocorrelations as the lag increases. If one is interested in using the generated values to obtain an estimate, then the variance of this estimate will be smaller the less correlated the generated values are. *Thinning* is the procedure when we only choose to store every k'th iteration, for $k \geq 2$. This can be done to get less correlated variables, but one then also discards a lot of values which again also contain information. Hence, this is not necessarily recommended. However, if we have restricted computer memory, thinning might be beneficial (Lesaffre and Lawson, 2012).

2.4.5 Cross-correlation plot

We can plot the chain values of one parameter against the values of another in a scatterplot to assess whether there seems to be correlation between the parameters. Thus, we plot $\theta_i^{(m)}$ against $\theta_j^{(m)}$ for all iterations. This is as mentioned in Lesaffre and Lawson (2012) a way to check for correlations among parameters if one for instance experiences convergence problems.

2.4.6 Histograms of marginal posterior distributions

The generated parameter values of the parameter θ_i from an MCMC run are besides from being generated from the posterior distribution also necessarily coming from its marginal posterior distribution (Chib and Greenberg, 1996). A natural part when analyzing the posterior distribution is to plot the histogram of these generated values (Ntzoufras, 2009). Recall that the posterior often includes many parameters, and hence it is much easier to look at one parameter at a time than at all at once.

2.5 Predictions based on MCMC output

After having run MCMC and discarded the burn-in period we get a series of samples from the posterior distribution. Let B iterations denote the burn-in period, and M the total number of iterations run. Let the generated values that we keep and use for prediction be denoted $\theta^{(B+1)}, ..., \theta^{M}$. If we are interested in predicting the value of a future observation $\tilde{x}$, we are interested in the poster predictive distribution given in (2.3). Draw $\tilde{x}^{(B+1)}, ..., \tilde{x}^{(M)}$ one at a time from the sampling distribution $f(x|\theta)$ by using the corresponding parameter generated from the posterior distribution, hence $\theta^{(B+1)}$ for the generation of $\tilde{x}^{(B+1)}$ and so on (Chib and Greenberg, 1996). Plot the predicted values as a histogram, and construct credible intervals to quantify the uncertainty in the predictions. For instance the 5%- and 95%-quantiles give an approximate 90% credible interval for the predicted value. The smaller the interval, the more sure are we about the corresponding predictions.

If one is rather interested in one value representing our best guess of the new value one can rather consider the posterior predictive mean. The estimated mean based on the predicted values is

$$\hat{E}[\tilde{x}|x] = \frac{1}{M-(B+1)} \sum_{m=B+1}^{M} \tilde{x}^{(m)}.$$

2.6 Probability distributions

To specify a model we need to specify the distribution of all variables on all levels in our model. There are infinitely many distributions to choose from, but it is often convenient to choose some standard distributions, whose parametric form and areas of application are well known. We present here briefly the distributions used in our model and MCMC algorithm, mainly so that there is no doubt about which parameterizations were used.

2.6.1 Poisson distribution

The Poisson distribution is a probability distribution for discrete random variables, and is used for counting data when one counts the number of events within a time interval or spatial region (Walpole et al., 2012). This can for instance be the number of telephone calls received by a help desk per hour, the number of machine failures per day or the number of typing errors per book page. Let Y be the number of occurrences within an interval or region. Let the intensity at which these events occur be denoted by $\lambda > 0$. If Y is Poisson distributed, which we denote $Y \sim \text{Poisson}(\lambda)$, then the corresponding probability mass function is given as

$$f(y|\lambda) = \frac{\lambda^y}{y!} e^{-\lambda}, \text{ for } y = 0, 1, 2, ... \tag{2.14}$$

The mean is $\text{E}[Y] = \lambda$ and is actually equal to the variance $\text{Var}[Y] = \lambda$. The full calculation of the mean is given in for instance Casella and Berger (2002). Note that this means that the intensity or rate λ is in fact the average rate of occurrence of events per unit time or space.

This distribution can in fact be derived from the following three basic assumptions (Walpole et al., 2012). i) The number of outcomes in one interval is independent of the number of outcomes in another disjoint interval. ii) For a short interval, the probability that exactly one outcome occurs is proportional to the length of the interval. iii) For a short interval, the probability of more than one outcome is so small it is neglected. These properties are also a guideline when considering the applicability of a Poisson distribution for the problem at hand.

One might also be interested in the distribution of the sum of occurrences due to different Poisson processes, which for instance can be the total amount of machines failures per hour when considering several machines. Let $Y_1, ..., Y_N$ denote independent random variables which are all Poisson distributed with corresponding intensities $\lambda_1, ..., \lambda_N$. Then the sum of these variables, hence $\sum_{i=1}^{N} Y_i$, is also Poisson distributed with intensity equal to the sum of the intensities $\sum_{i=1}^{N} \lambda_i$. The derivation of this result can be found in Larsen and Marx (2018).

2.6.2 Gamma distribution

Let $a > 0$ be a shape parameter and $b > 0$ an inverse scale or rate parameter in a gamma distribution. If the continuous variable Y is gamma distributed, hence $Y \sim \mathrm{G}(a, b)$, then its corresponding probability density function is given as

$$f(y|a,b) = \frac{b^a}{\Gamma(a)} y^{a-1} e^{-by}, \text{ for } y > 0 \tag{2.15}$$

where $\Gamma(a)$ is the gamma function (Walpole et al., 2012). This function is defined as

$$\Gamma(a) = \int_0^\infty u^{\alpha-1} e^{-u} du, \text{ for } a > 0.$$

Note that for a positive integer n the gamma function can rather be expressed as

$$\Gamma(n) = (n-1)!$$

where $n = 1, 2, ...$

The mean of the gamma distribution is $\mathrm{E}[Y] = \frac{a}{b}$ and the variance is $\mathrm{Var}[Y] = \frac{a}{b^2}$.

2.6.3 Inverse-gamma distribution

Let $a > 0$ denote a shape parameter and $b > 0$ a scale parameter of an inverse-gamma distribution. If $Y \sim \mathrm{IG}(a, b)$ then the probability density function as listed in (Gelman et al., 2014) is

$$f(y|a,b) = \frac{b^a}{\Gamma(a) y^{(a+1)}} e^{-\frac{b}{y}}, y > 0 \tag{2.16}$$

with corresponding mean $\mathrm{E}[Y] = \frac{b}{a-1}$ if $a > 1$ and variance $\mathrm{Var}[Y] = \frac{b^2}{(a-1)^2(a-2)}$ if $a > 2$.

Note that if Y is Gamma distributed, hence $Y \sim \mathrm{G}(a, b)$ with the parameterization as in (2.15), then $\frac{1}{Y}$ is inverse-gamma distributed with shape parameter a and scale b. This is easily seen using the standard transformation formula.

2.6.4 Exponential distribution

Let Y be a continuous random variable, which can only take on non-negative values. If it is exponentially distributed we denote it as $Y \sim \mathrm{Exp}(\lambda)$, where $\lambda > 0$ is the intensity parameter. The corresponding probability density function is

$$f(y|\lambda) = \lambda e^{-\lambda y}, \text{ for } y > 0.$$

Here the corresponding mean and variance are $\mathrm{E}[Y] = \frac{1}{\lambda}$ and $\mathrm{Var}[Y] = \frac{1}{\lambda^2}$, respectively.

The exponential distribution is in fact a special case of the aforementioned gamma distribution, where the scale parameter of the gamma distribution is set equal to one

(Walpole et al., 2012). If rather parameterized by its mean $m = \mathrm{E}[Y]$ we get the following expression for the probability density function

$$f(y|m) = \frac{1}{m} e^{-\frac{1}{m}y}, \text{ for } y > 0, m > 0. \tag{2.17}$$

Then we naturally have $\mathrm{E}[Y] = m$ and $\mathrm{Var}[Y] = m^2$.

2.6.5 Normal distribution

Let Y be normally distributed with mean $\mu \in (-\infty, \infty)$ and variance $\sigma^2 > 0$. Then the density of Y is (Walpole et al., 2012)

$$f(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}. \tag{2.18}$$

The corresponding cumulative distribution function is then

$$\begin{aligned} F(y) = P(Y \leq y) &= \int_{-\infty}^{y} f(u|\mu, \sigma^2) du \\ &= \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(u-\mu)^2}{2\sigma^2}} du. \end{aligned}$$

In the case when $\mu = 0$ and $\sigma = 1$ we have the standard normal distribution and we typically denote its probability density function and cumulative distribution function by ϕ and Φ, respectively. Recall how the cumulative distribution function for $Y \sim \mathrm{N}(\mu, \sigma^2)$ can be expressed by the use of the cumulative distribution of the standard normal as follows

$$F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right).$$

If rather parameterized with the variance $v = \sigma^2$ we get the following probability density function

$$f(y|\mu, v) = \frac{1}{\sqrt{2\pi v}} e^{-\frac{(y-\mu)^2}{2v}}. \tag{2.19}$$

2.6.6 Lognormal distribution

If Y is lognormally distributed, and hence $Y \sim \mathrm{lognormal}(\mu, \sigma^2)$, it means that $\ln Y$ is normally distributed with mean μ and standard deviation σ (Walpole et al., 2012). Naturally Y must be positive-valued since the logarithm is only defined for positive values. The corresponding probability density function for Y is then

$$f(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma \cdot y} e^{-\frac{(\ln y - \mu)^2}{2\sigma^2}}, \text{ where } \mu \in \mathbb{R}, \sigma > 0, y > 0.$$

The corresponding mean is $\mathrm{E}[Y] = e^{\mu+\frac{1}{2}\sigma^2}$ and the variance is given by $\mathrm{Var}[Y] = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1)$.

The cumulative distribution function for Y is

$$F(y) = P(Y \leq y) = \int_0^y \frac{1}{\sqrt{2\pi}\sigma \cdot u} e^{-\frac{(\ln u - \mu)^2}{2\sigma^2}} du. \tag{2.20}$$

Since $\ln Y \sim \mathrm{N}(\mu, \sigma^2)$ we can rather express (2.20) as

$$\Phi\left(\frac{\ln y - \mu}{\sigma}\right).$$

If interested in the quantity $\ln(1 - F(y))$ this can be computed as

$$\ln(1 - F(y)) = \ln\left(\Phi\left(\frac{-(\ln y - \mu)}{\sigma}\right)\right) \tag{2.21}$$

by noting that in general we have

$$\Phi(-y) = 1 - \Phi(y)$$

for the cumulative distribution of the standard normal.

2.6.7 Continuous uniform distribution

Let Y be a continuous random variable which has equal probability of being at either point in the fixed interval [a,b], where $a, b \in \mathbb{R}$ and $a < b$. Then its corresponding density function is given as

$$f(Y|a, b) = \frac{1}{b-a}, \text{ for } a \leq y \leq b.$$

Note that the interval does not necessarily need to be closed, as the same distribution applies for the open interval (a, b) (Walpole et al., 2012).

3 Data

We incorporate wind speed in our model through what we denote *wind exposure*, which is a function of wind speed and segment length. The segment length and wind speed for each segment are treated as known and given, and these are what one often refers to as explanatory variables. When we consider the number of failures in one hour for a segment it is then conditioned on the segment length and wind speed along that segment for the same hour. The number of failures on the other hand is treated as unknown and random. To be able to use our model for inference we therefore need to know the length of all power line segments and have a data set of wind speeds in addition to observed failures for the same hours. This data is provided by Statnett, for nine power lines in total, all located in Northern Norway. We let the power lines be anonymous and denote them by the numbers 1 to 9. In this section we present summaries of the data sets to give some insight in typical values and their range.

3.1 Wind data

The only information about wind that we include in our model, and which Statnett is already using for their own model, is the wind speed measured in [m/s]. Kjeller Vindteknikk has provided a reanalysis data set for internal use at Statnett, consisting of historical hourly time series for several weather parameters in Norway, including wind speed at different heights. We consider only what corresponds to the wind speed at a height of 18 metres above ground level. These time series are based on several simulations from a numerical weather prediction model called WRF (Weather Research and Forecasting model). For wind speed the corresponding grid size is 1km $\times$ 1km. We have available reanalysis data for nine different overhead transmission lines on an hourly basis in a period from January 1, 1998, 00:00 up to and including February 28, 2015, 23:00. Note that these times are according to UTC (Coordinated Universal Time). To map weather data from the weather grid to the power lines, the geographical position of the power towers are used. One then treats the wind at the location in the grid nearest a power tower as the wind speed representative for a whole segment, namely for the upcoming segment. Note that since the segment of power lines often are far shorter than the grid size, one might get the same wind speed for several segments as they all are associated with the same grid point in the reanalysis data set. The wind speeds are available for each whole hour, and are to be interpreted as the mean wind speed for the previous hour. Hence, the wind at 12:00 is then to be interpreted as the wind representative for the hour 11:00-12:00.

In the model presented in Solheim et al. (2016) one has set a threshold value of 15 m/s such that there is a non-zero probability of failure only when the wind is above this threshold. It is therefore of interest to know approximately how often this threshold is exceeded. For each segment we let the number of hours for which the wind speed for this segment is above the wind threshold divided by the total amount of hours of data available be an estimate of the probability of exceeding the wind threshold. We get one such estimate for each segment and grouped line by line these are displayed through the histograms in Figure 5. The same plots based on only six months of weather data from January-June, 2014, are shown in Figure 6.

As seen from the histograms in Figure 5 the typical values are below 0.020, which indicates that most of the time the wind along a segment is below 15 m/s. In addition, for most lines there are fewer and fewer segments the larger the estimate value as seen by the decrease in bar heights along the horizontal axis.

3.2 Segment lengths

We need the length of each line segment for the calculation of wind exposure used in the model. These lengths are given in metres. Preferable a segment length should also be positive and non-zero, as it makes little sense to talk of a line of zero length. Since the data set of segment lengths had some inconsistencies in the naming of the power towers within a line, between lines and compared to the wind data, we needed to do some pre-processing of it. First of all we needed to order the lengths such that the first length corresponded to the actual first segment of a line relative to its starting point,

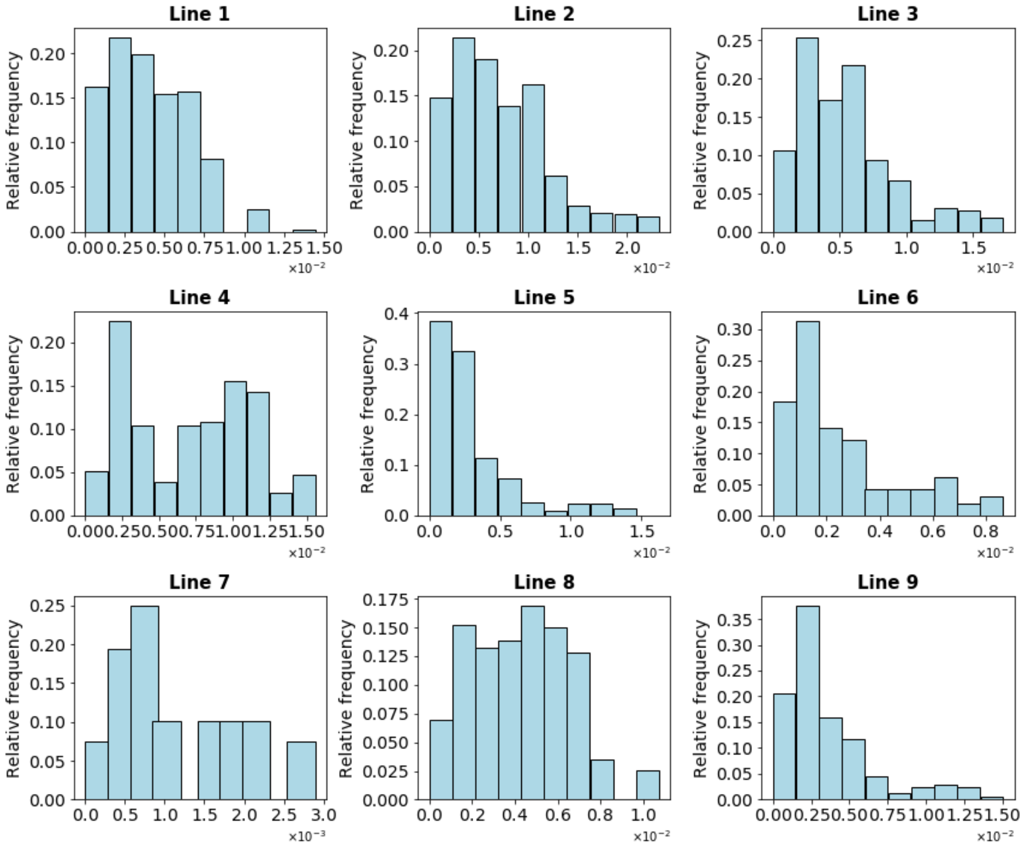

Figure 5: Histograms of each segment's estimated probability of having a corresponding wind speed above the threshold value of 15 m/s based on all available wind data. These estimates are grouped by line.

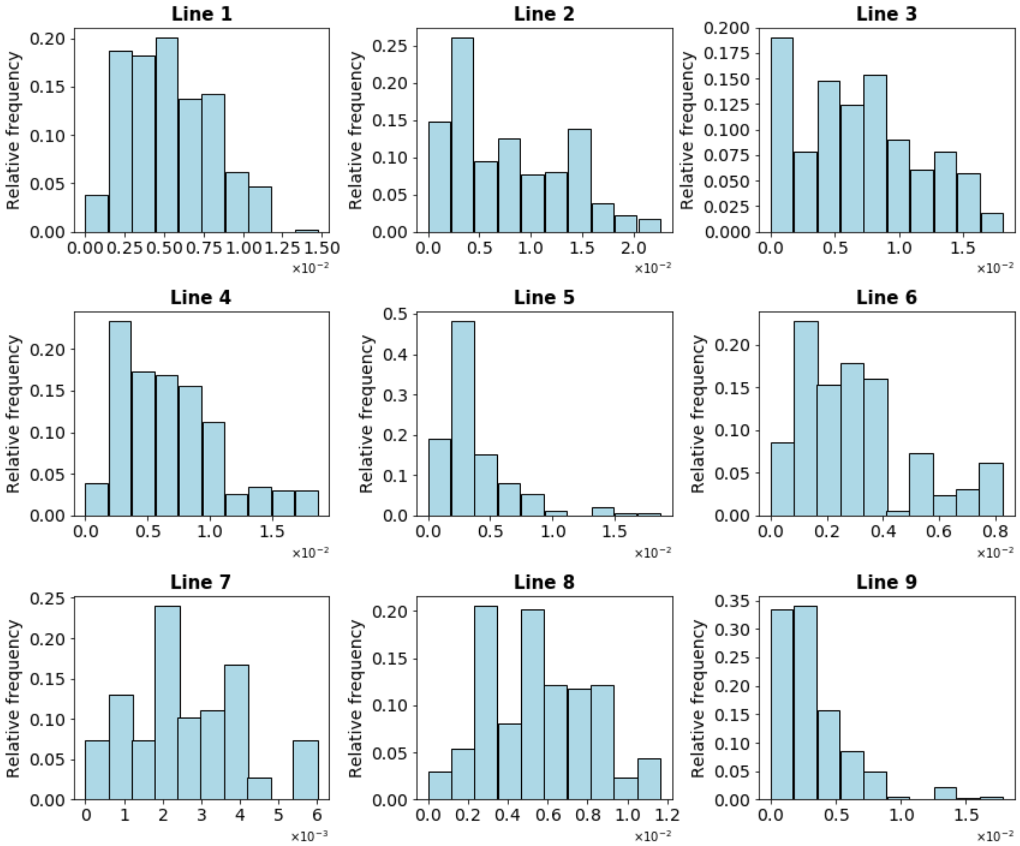

Figure 6: Histograms of each segment's estimated probability of having a corresponding wind speed above the threshold value of 15 m/s based on wind data from January-June, 2014. The estimates are grouped by line.

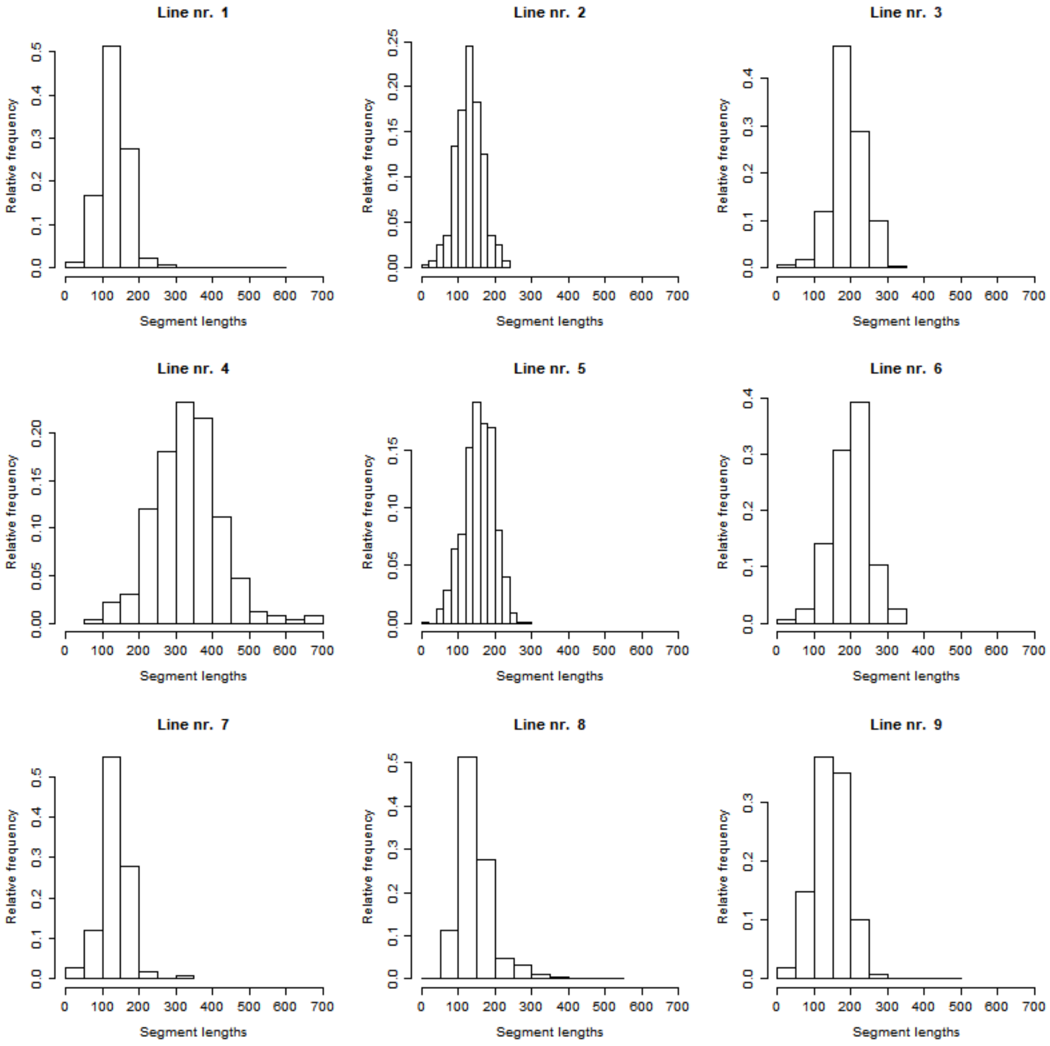

Figure 7: Histograms of the segment lengths in metres for each overhead transmission line consider in this report.

the second segment to its actual second line segment and so forth. For eight out of the nine lines there was also a segment length of zero or NaN (not a number). We changed these values to be equal to 1. This was a somewhat arbitrary choice, but since these were typically the last segment length of the power line, and hence for the segment going into a station for instance, they are usually not that long. As we are unsure of their actual value we want to avoid that they influence the results too much, hence we let they have such a short length. For the lines we consider in this report they consist of on average 431 segments each. The segment lengths are displayed in the histograms seen in Figure 7, one for each line. From the histograms we see that the segment lengths are usually no longer than a few hundred metres. However the lengths of segments within a line can vary some.

Line	# failures, max $u^t_{n,l} > u_{\text{threshold}}$	# failures, max $u^t_{n,l} \leq u_{\text{threshold}}$
1	3	0
2	0	0
3	1	2
4	17	0
5	0	0
6	0	0
7	0	1
8	0	0
9	0	0

Table 1: Table of temporary wind failures for all lines in the time period January 1, 1998, 00:00 (UTC) up to and including February 28, 2015, 23:00 (UTC). The failures are divided into tow groups based on whether the corresponding maximum wind speed along that line at the time of failure, denoted max $u^t_{n,l}$ was above or below the wind threshold $u_{\text{threshold}} = 15$ m/s.

3.3 Failure data

In this report we only consider temporary failures on overhead transmission lines due to wind, and hence only a part of the total amount of failures reported. These type of failures are most common in Norway during the winter months. The failures are reported line-wise, and not segment-wise. Hence, we have neither information of exactly where nor approximately where along the line a failure occurred. The exact time is given for each failure, which is reported in local time, i.e. UTC+1 for winter time and UTC+2 for summer time. We have available failure data for the same time period as for the wind data. There have been 24 failures in total for the given time period. The number of failures for each line, grouped by whether the failure occurred in an hour where the maximum wind speed along the corresponding line was above the threshold or not, is displayed in Table 1. Note that if the maximum wind speed along a line is above the threshold, this means that at least one segment has a wind speed of more than 15 m/s. As is evident from Table 1 the majority of lines have no temporary wind failures in the given time period, however this does not mean that these lines in general have zero probability of failure.

4 Formulate models and simulate failure data

Having established the theory of Bayesian hierarchical models and introduced the data available we now turn to the actual formulation of our models. The model for failures is depicted as a directed acyclic graph (DAG) and explained in a bottom-up fashion starting with the data level and proceeding with the parameters, first at a prior level and then at a hyperprior level. Since we have so to say no knowledge of in which range the parameters should lie, and since the parameters are hard to give any interpretation, we use uninformative priors on the top level. We then introduce how we simulated our own data set, which was used for MCMC. One could have, and ideally would want to, use the actual observed failure data. However, since having limited time and computational power available we decided it was better to simulate our own failure data for the MCMC runs.

4.1 Our two Bayesian hierarchical models

We formulate two Bayesian hierarchical models for which the only difference is in terms of the likelihood. This means that the parameters included, in addition to the corresponding prior and hyperprior distributions, are the same. The failure data available to us only contains failures reported line-wise, and hence we have no information of which segment or segments that failed. This is also the type of data used in Solheim et al. (2016) and we therefore find it natural to start by formulating a model based on the same type of data. However, since the wind data is given per segment and since the line-wise model is based on independence between segments we introduce a segment-wise model as well. This model is analogue to the line-level one, only that we here require failure data to be reported segment-wise. Such data is not available to us at the moment so we simulate our own data set. However, one might imagine that this type of failure data is possible to obtain in the future if one requires the reporting of failures to be somewhat more detailed. We base the following models on concepts and assumptions from the model presented in Solheim et al. (2016).

4.1.1 Line-wise model

The overall hierarchical structure of our model is illustrated by the DAG in Figure 8. Note that we have used Greek letters for the middle layer, hence the prior level, and Roman letters for the data layer and for the hyperprior level of our model. This is to clearly distinct the different parts of the model. We start by introducing the data level, which corresponds to the grey nodes in the DAG. Note that these are the only nodes colored grey, which means that these are the only observable variables.

Let $l = 1, ..., L$ denote overhead transmission lines. In our case we look at only a few of the power lines operated by Statnett, nine in total, hence $L = 9$. We let a specific hour be denoted by t, and such that $t = 1, ..., T$ are all the hours for which we have wind data. Further we let the number of failures along line l within hour t be denoted by x_l^t. We assume the failures to follow a Poisson process with corresponding intensity λ_l^t. As mentioned in Section 2 this implies that we assume failures on a line within an hour to occur independent of each other, and that only one failure can occur at a time.

So we have

$$x_l^t \sim \text{Poisson}(\lambda_l^t),$$

but need a way to connect wind, or more precisely wind exposure, to the failure rate. Since having wind data on segment-level we decide to look at a line as a series system consisting of N_l segments, and where the number of failures per hour for each segment is Poisson distributed. Let a segment-wise Poisson process have intensity $\lambda_{n,l}^t$, with n being the index for a particular segment. These processes are assumed independent,

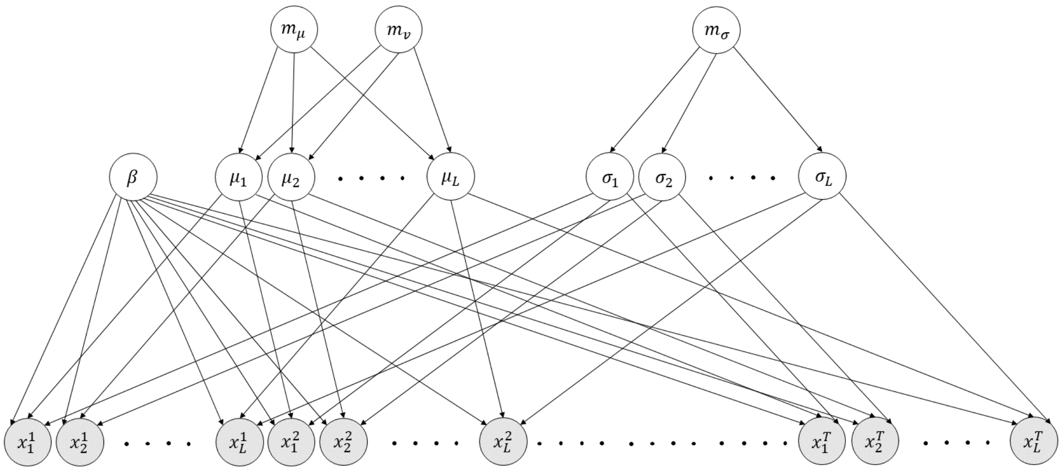

Figure 8: Visualization of the line-wise Bayesian hierarchical model as a DAG. The grey nodes represent the data level, the mid level is the prior level and the top level is the hyperprior level. The nodes all represents random variables, where the bottom level contains the only observable ones. Therefore these are colored grey. The arrows pointing towards a node show which parameters that are part of the corresponding probability density or probability mass function of the node they are pointing towards.

which implies that we assume the amount of failures for one segment to be unaffected by the number of failures for other segments. The total number of failures for a line per hour is the sum of the failures along each segment within the same time frame, and we therefore have

$$x_l^t = \sum_{n=1}^{N_l} x_{n,l}^t,$$

where $n = 1, ..., N_l$ denotes the different segments for line l. Recall from the theory for the Poisson distribution that the sum of independent Poisson distributed random variables is again Poisson distributed with intensity given as the sum of each one's intensity. The distribution of total failures is then also Poisson with intensity

$$\lambda_l^t = \sum_{n=1}^{N_l} \lambda_{n,l}^t. \tag{4.1}$$

The corresponding probability mass function $f(x_l^t|\lambda_l^t)$ is given as in (2.14).

Since we are looking at failures due to wind, we assume wind speed to be the main explanatory variable. We want to connect the probability of failure to the wind speed. For this we us wind exposure, which is a function of wind speed cubed and segment length. The concept of wind exposure is described in the article Solheim et al. (2016), but we have chosen to make some changes to the original definition. Recall that l denotes a specific transmission line, consisting of N_l line segments. The length of segment n for line l is $d_{n,l} > 0$. Further we let $u_{n,l}^t$ be the wind speed in [m/s] along segment n for line l in hour t. Most wind failures naturally do happen when the wind is strong and one has therefore decided to set a threshold of $u_{\text{threshold}} = 15$ m/s in the original definition of wind exposure. Then only wind speeds above this threshold contribute to the probability of failure. We keep the same threshold value, however we avoid any probabilities to be strictly zero.

The expression for wind exposure used in this report, which we denote $w_{n,l}^t$ for the wind exposure for segment n, is given as

$$w_{n,l}^t(\beta) = \begin{cases} d_{n,l}\left(\alpha\left(u_{n,l}^t - u_{\text{thres}}\right)^3 + \beta\right) & \text{for } u_{n,l}^t > u_{\text{thres}}, \\ d_{n,l}\beta & \text{for } u_{n,l}^t \leq u_{\text{thres}}, \end{cases}$$

where $\alpha > 0$ is a known scaling parameter and $\beta > 0$ an unknown constant. To make wind exposure unitless α must necessarily be given in $[\frac{\text{s}^3}{\text{m}^4}]$ and β in $[\frac{1}{\text{m}}]$. Since we have no further knowledge of the value of α we assign it a fixed value of 1 and omit it in the rest of the expressions to not confuse it with a random variable. The segment length $d_{n,l}$ and wind speed $u_{n,l}$ are both assumed known. The wind exposure is then

$$w_{n,l}^t(\beta) = d_{n,l}\left(\left(u_{n,l}^t - u_{\text{thres}}\right)_+^3 + \beta\right), \tag{4.2}$$

where we let

$$\left(u_{n,l}^t - u_{\text{thres}}\right)_+ = \begin{cases} (u_{n,l}^t - u_{\text{thres}}) & \text{for } u_{n,l}^t > u_{\text{thres}}, \\ 0 & \text{for } u_{n,l}^t \leq u_{\text{thres}}. \end{cases}$$

By introducing a constant $\beta > 0$ in this way we avoid the possibility that wind exposure can be strictly equal to zero. As we soon explain, this would have given a probability of zero for failure along a segment, which is not desired in our case. Even though we altered the expression for wind exposure slightly we kept the linear relationship with respect to segment length. By this we mean that if one plots wind exposure against segment length, while keeping everything else fixed, it yields a linear graph. This is the case for the model described in Solheim et al. (2016) when the wind is above the threshold, otherwise their definition yields wind exposure equal to zero for all lengths. With our definition the wind exposure is higher the longer the segment length is, both when the wind speed is above and below the wind threshold.

Let $p_{n,l}^t$ be the probability of at least one temporary failure due to wind on segment n for line l in hour t. In addition, let μ_l and σ_l be the parameters of a lognormal distribution with corresponding cumulative probability distribution F given by (2.20). As in the article Solheim et al. (2016) we are going to relate the probability of any failures to wind exposure as follows

$$p_{n,l}^t = p_{n,l}^t(w_{n,l}^t(\beta), \mu_l, \sigma_l) = F(w_{n,l}^t(\beta)|\mu_l, \sigma_l) \tag{4.3}$$

where the notation $p_{n,l}^t(w_{n,l}^t(\beta), \mu_l, \sigma_l)$ stresses the fact that this probability is a function of both wind exposure and line-specific parameters μ_l and σ_l. However, we refer to this only as $p_{n,l}^t$ from now on. The cumulative function in (4.3) is only strictly equal to zero for a wind exposure equal to zero, however as our definition of wind exposure from (4.2) ensures non-zero and positive values for the wind exposure, the probability of failure is also ensured non-zero. Note that the parameters μ_l and σ_l in the lognormal distribution are not time-dependent. Recall that we assumed a Poisson distribution for the number of failures along a line segment, hence we have that

$$\begin{aligned} p_{n,l}^t &= P(\text{at least one failure on line } l\text{, segment } n\text{, in hour } t) \\ &= 1 - P(\text{no failure on line } l\text{, segment } n\text{, hour } t) \\ &= 1 - f(x_{n,l}^t = 0|\lambda_{n,l}^t) \\ &= 1 - e^{-\lambda_{n,l}^t} \end{aligned} \tag{4.4}$$

by using the probability mass function given by (2.14). When solved for the intensity as a function of $p_{n,l}^t$ we obtain

$$\lambda_{n,l}^t = -\ln(1 - p_{n,l}^t). \tag{4.5}$$

We then easily find the intensity parameter corresponding to the Poisson distribution for the total amount of failures x_l^t along a line per hour. Inserting the expression for the segment-wise intensity (4.5) in the expression for the line-wise intensity given by (4.1) yields

$$\lambda_l^t = \sum_{n=1}^{N_l} -\ln(1 - p_{n,l}^t) = \sum_{n=1}^{N_l} -\ln(1 - F(w_{n,l}^t(\beta)|\mu_l, \sigma_l)). \tag{4.6}$$

Note that we inserted (4.3) for the probability of any segment failure, hence it is seen that the line-wise intensity is indeed a function of the unknown parameters β, μ_l and σ_l. One could then rather use a notation of the form $\lambda_l^t(\beta, \mu_l, \sigma_l)$. However, we choose to continue with the simplest notation. The probability for at least one failure along a line, which we denote p_l^t, is found in the same manner as in (4.4) to be

$$\begin{aligned} p_l^t &= 1 - f(x_l^t = 0|\lambda_l^t) \\ &= 1 - e^{-\lambda_l^t} \\ &= 1 - e^{-\sum_{n=1}^{N_l} \lambda_{n,l}^t}. \end{aligned} \tag{4.7}$$

Let $\boldsymbol{x}^t = (x_1^t, ..., x_L^t)$ be the vector containing all random variables representing the number of failurs in hour t along each of the L lines. The corresponding rates of failures are denoted $\boldsymbol{\lambda}^t = (\lambda_1^t, ..., \lambda_L^t)$. We assume all x_l^t conditional independent given λ_l^t, hence the distribution of x_l^t is not dependent on the number of failures on any other lines, and not even on the number of failures for any other hours. The likelihood, which represents the distribution for the data-level in our model, is then equal to

$$\begin{aligned} &f(\boldsymbol{x}^1, ..., \boldsymbol{x}^T|\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^T) \\ &= \prod_{t=1}^{T}\prod_{l=1}^{L} f(x_l^t|\lambda_l^t) \\ &= \prod_{t=1}^{T}\prod_{l=1}^{L} \frac{(\sum_{n=1}^{N_l} -\ln(1 - F(w_{n,l}^t(\beta)|\mu_l, \sigma_l)))^{x_l^t} \cdot e^{-\left(\sum_{n=1}^{N_l} -\ln\left(1-F(w_{n,l}^t(\beta)|\mu_l,\sigma_l)\right)\right)}}{(x_l^t)!} \end{aligned}$$

where the intensity λ_l^t has the expression as in (4.6).

We proceed with the prior level, hence the mid-level in Figure 8. As mentioned in the explanation of wind exposure we let the unknown constant β be a positive random variable. Since we have no further knowledge of this we assign it an uninformative prior, chosen as the improper uniform distribution on the whole positive real line. Hence

$$f(\beta) \propto 1 \cdot I_{\beta>0}(\beta)$$

where we let the general expression for the inidicator function $I_A(y)$ be as follows

$$I_A(y) = \begin{cases} 1 \text{ for } y \in A, \\ 0 \text{ otherwise.} \end{cases}$$

Note that we treat β as a common parameter for all lines. We believe this to be suitable as the probablity of failure has the opportunity to vary from line to line due to the line-specific parameters instead.

For the parameter μ_l that appears in the cumulative function of the lognormal we assign it a normal prior distribution. Recall that this parameter can be seen as the mean in a normal distribution, and hence $\mu_l \in \mathbb{R}$. The normal distribution has the same support and is therefore a possible and appropriate choice as prior. Let $m_\mu \in \mathbb{R}$ denote the mean in its prior distribution and $v_\mu > 0$ the variance. Thus

$$\mu_l | m_\mu, v_\mu \sim N(m_\mu, v_\mu) \text{ for } l = 1, ..., L$$

where the corresponding probability density function is given by (2.19).

The other parameter σ_l in the cumulative lognormal can be viewed as the corresponding standard deviation in a normal distribution. We assign it an exponential prior distribution since this distribution has a positive support, which is the same support needed as necessarily $\sigma_l > 0$. Due to this choice we also avoid a heavy tail as the exponential function decays fast. Let the mean of the prior distribution be denoted by m_σ and we then have

$$\sigma_l | m_\sigma \sim \text{Exp}(m_\sigma) \text{ for } l = 1, ..., L.$$

The corresponding density is given in (2.17). In addition we assume the μ_l's to be conditional independent, i.e. that $\mu_l | m_\mu, v_\mu$ is independent of $\mu_j | m_\mu, v_\mu$ for $l \neq j$. Corresponding assumptions are made for $\sigma_1, ..., \sigma_L$, such that $\sigma_l | m_\sigma, v_\sigma$ and $\sigma_j | m_\sigma, v_\sigma$ are independent for $l \neq j$.

Next we present the hyperprior distributions, and hence the top level of our model. The random variables on this level are the parameters in the prior distributions, i.e. the mean and variance of the normal prior distribution for μ_l and the mean of the exponential prior of σ_l. We assign improper and uninformative distributions to all these parameters. This choice is based on the fact that we per now have no information on what typical values might be and that these parameters are hard to interpret directly as they influence parameters that are part of intensity parameters that again affect the probability of failure. Besides, by letting they have uninformative distributions we also ensure that the posterior distribution is mostly influenced by the likelihood.

We let $m_\mu \in \mathbb{R}$ have an improper uniform distribution on the whole real line, such that

$$f(m_\mu) \propto 1. \tag{4.8}$$

For the variance parameter we also assume an improper prior distribution. Since $v_\mu > 0$ we choose an uninformative distribution on the whole positive part of the real line, and as mentioned in the theory part one option is

$$f(v_\mu) \propto \frac{1}{v_\mu} \cdot I_{v_\mu > 0}(v_\mu). \tag{4.9}$$

In the same manner we assign an improper prior distribution for the mean m_σ of the exponential distribution, namely the improper uniform distribution restricted on the positive part of the real line. We then have

$$f(m_\sigma) \propto 1 \cdot I_{m_\sigma > 0}(m_\sigma). \tag{4.10}$$

In addition we assume m_μ, v_μ and m_σ to be independent.

Lastly, we summarize our model. The parameters which we treat as unknown are $\beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu$ and m_σ. In our case this corresponds to 22 parameters in total when considering $L = 9$ power lines. As mentioned in the theory section the distribution we are interested in is the posterior distribution. Recall from (2.2) that the posterior distribution is proportional to the full joint distribution. Let $\boldsymbol{x}$ denote all variables representing number of line-wise failures along all lines for all hours available and $\boldsymbol{\mu}$ as all μ_l's such that $\boldsymbol{\mu} = (\mu_1, .., \mu_L)$. Analogously we let $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_L)$. We summarize the model by writing out the joint distribution as follows

$$\begin{aligned} f(\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}, m_\mu, v_\mu, m_\sigma | \boldsymbol{x}) &\propto f(\boldsymbol{x}, \beta, \boldsymbol{\mu}, \boldsymbol{\sigma}, m_\mu, v_\mu, m_\sigma) \\ &\propto f(\boldsymbol{x}|\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}) f(\boldsymbol{\mu}|m_\mu, v_\mu) f(\boldsymbol{\sigma}|m_\sigma) f(m_\mu) f(v_\mu) f(m_\sigma) f(\beta) \\ &\propto \left[\prod_{t=1}^{T} \prod_{l=1}^{L} f(x_l^t | \lambda_l^t(\beta, \mu_l, \sigma_l))\right] \cdot \left[\prod_{l=1}^{L} f(\mu_l | m_\mu, v_\mu)\right] \\ &\quad \cdot \left[\prod_{l=1}^{L} f(\sigma_l | m_\sigma, v_\sigma)\right] \cdot f(\beta) f(m_\mu) f(v_\mu) f(m_\sigma) \end{aligned} \tag{4.11}$$

where we have used rules for conditioning and the assumption that all variables are conditional independent of all other variables given their parent nodes in the DAG from Figure 8. So for instance $\boldsymbol{\mu}$ is independent of $\beta, \boldsymbol{\sigma}, m_\sigma$ given m_μ and v_μ. Recall that the parameters on the hyperprior level are assumed independent, in addition we assume them independent from β.

4.1.2 Segment-wise model

We adjust the aforementioned model to suit segment-wise observed failures. Recall that we have no observed data for segment-wise failures, but one can imagine that one could improve the collection of failure data in the future as to report more specific about where along a line a failure occurred. We let our segment-wise model have the same parameters and distributions for these as presented earlier in this section. The only adjustment required is a change of the expression for the likelihood. Let $x_{n,l}^t$ denote the number of failures along segment n at line l for hour t with corresponding intensity $\lambda_{n,l}^t$ given by (4.5). Let $\boldsymbol{x}_l^t = (x_{1,l}^t, ..., x_{N_l,l}^t)$ denote all the segment-wise failures along line l in hour t with corresponding intensities $\boldsymbol{\lambda}_l^t = (\lambda_{1,l}^t, ..., \lambda_{N_l,l}^t)$. Again assuming all segments, lines and hours independent, the likelihood in our segment-wise model is

$$\begin{aligned}
&f(\boldsymbol{x}_1^1, ..., \boldsymbol{x}_L^1, ..., \boldsymbol{x}_1^T, ..., \boldsymbol{x}_L^T | \boldsymbol{\lambda}_1^1, ..., \boldsymbol{\lambda}_L^1, ..., \boldsymbol{\lambda}_1^T, ..., \boldsymbol{\lambda}_L^T) \\
&= \prod_{t=1}^{T} \prod_{l=1}^{L} \prod_{n=1}^{N_l} f(x_{n,l}^t | \lambda_{n,l}^t) \\
&= \prod_{t=1}^{T} \prod_{l=1}^{L} \prod_{n=1}^{N_l} \frac{e^{-\left(-\ln\left(1-F(w_{n,l}^t(\beta)|\mu_l,\sigma_l)\right)\right)} (-\ln(1-F(w_{n,l}^t(\beta)|\mu_l,\sigma_l)))^{x_{n,l}^t}}{(x_{n,l}^t)!} \\
&= \prod_{t=1}^{T} \prod_{l=1}^{L} \frac{e^{-\left(\sum_{n=1}^{N_l} -\ln\left(1-F(w_{n,l}^t(\beta)|\mu_l,\sigma_l)\right)\right)} \prod_{n=1}^{N_l} \left[(-\ln(1-F(w_{n,l}^t(\beta)|\mu_l,\sigma_l)))^{x_{n,l}^t}\right]}{(x_{1,l}^t)! \cdots (x_{N_l,l}^t)!}
\end{aligned} \tag{4.12}$$

where the last line is obtained by incorporating all segment-wise factors for a line.

The expression for posterior distribution is obtained by inserting the segment-wise likelihood (4.12) instead of the line-wise in (4.11). This yields

$$\begin{aligned}
f(\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}, m_\mu, v_\mu, m_\sigma | \boldsymbol{x}) \propto &\left[\prod_{t=1}^{T} \prod_{l=1}^{L} \prod_{n=1}^{N_l} f(x_{n,l}^t | \lambda_l^t(\beta, \mu_l, \sigma_l))\right] \cdot \left[\prod_{l=1}^{L} f(\mu_l | m_\mu, v_\mu)\right] \\
&\cdot \left[\prod_{l=1}^{L} f(\sigma_l | m_\sigma, v_\sigma)\right] \cdot f(\beta) f(m_\mu) f(v_\mu) f(m_\sigma)
\end{aligned} \tag{4.13}$$

where $\boldsymbol{x}$ now denotes all segment-wise number of failures for all lines and all hours.

4.2 Simulated data

Ideally we would like to use all or major parts of the failure data available, but as we have limited computational resources available we decide to limit ourselves to six months of data. Because of this we are able to produce some results in a reasonable amount of time. To account for the difference in weather between winter and summer months, and thereby also in the amount of failures reported in different seasons, we decided to look at the period January-June, 2014. The year 2014 was chosen somewhat arbitrarily, however we know there has been reported failures during this year. For the rest of the report we only consider the hours of wind data for January 1, 2014, 00:00 (UTC) - June 30, 2014, 23:00 (UTC).

Since there are few failures observed in general, which is also the case in our chosen time period, we choose to rather simulate failure data. This choice also enables us to later compare results with some true parameter values, and hence gives a natural way to evaluate the models. Recall that we have failure data from a period of about 17 years, or more precisely from January 1, 1998, to February 28, 2015. For this period there are 24 failures in total considering all lines. We simulate data such that the total amount of failures within our chosen six months is approximately the same number as the number of failures as we have for the whole time period that we have failure data for. In addition we want to be sure that the simulated data somehow mimics the true, and hence so that we do get some failures at times where the wind is also below the threshold value. This is indeed the case in the observed data set, as seen from Table 1.

To be able to simulate failure data we need to set values for the parameters that are in general treated as unknown. We started simply by setting $\beta = 1$. Combining wind data and this value of β the wind exposure for each segment for each hour were calculated. By finding the max value for the wind exposure for line l in the period January-June, 2014, which we denote max $w_{n,l}^t$, we plot the intensity (4.6) as a function of wind exposure for the range $[0, \max\, w_{n,l}^t]$, see Figure 9. The wind exposure is a function of wind cubed so we also plot the intensity against wind speed in Figure 10 since we are more familiar with values of wind speed than values of wind speed cubed. This is done for several different parameter values for μ_l and σ_l. We believe that the intensity should either increase for increased wind speeds, and that the increase should be larger for higher than lower values, or increase steadily, hence show a linear trend. At least we try to avoid a shape that increases rapidly for small values, since this would give a probability of failure that hardly changes for an increase in higher wind speeds. By visual inspection we chose values for the parameters μ_l and σ_l so that the shape of the aforementioned intensity plots looked as we wanted them to look, and so that the simulated failures were spread across several lines. In addition, we maintained a decent number of total failures and a similar proportion of failures with corresponding wind above or below the threshold as for the actual failure data.

For the simulated data we used the following parameter values

$$\begin{aligned}
&\beta = 1 \\
&\boldsymbol{\mu} = \{16.5, 20.0, 23.5, 18.0, 24.1, 19.2, 28.5, 15.5, 22.0\} \\
&\boldsymbol{\sigma} = \{2.0, 3.2, 3.8, 2.0, 3.0, 2.8, 3.5, 1.9, 1.1\}
\end{aligned} \tag{4.14}$$

which gave a simulated data set consisting of 26 failures in total. These are spread across four of the lines, see Table 2 where we have separated failures that occur based on whether the corresponding wind speed for the segment that failed was above or below the threshold of 15 m/s. We use the same failure data for the line-wise and segment-wise model, i.e. the only difference being that for line-wise failures we aggregate the simulated data to only consist of the total number of failures along a line within a given

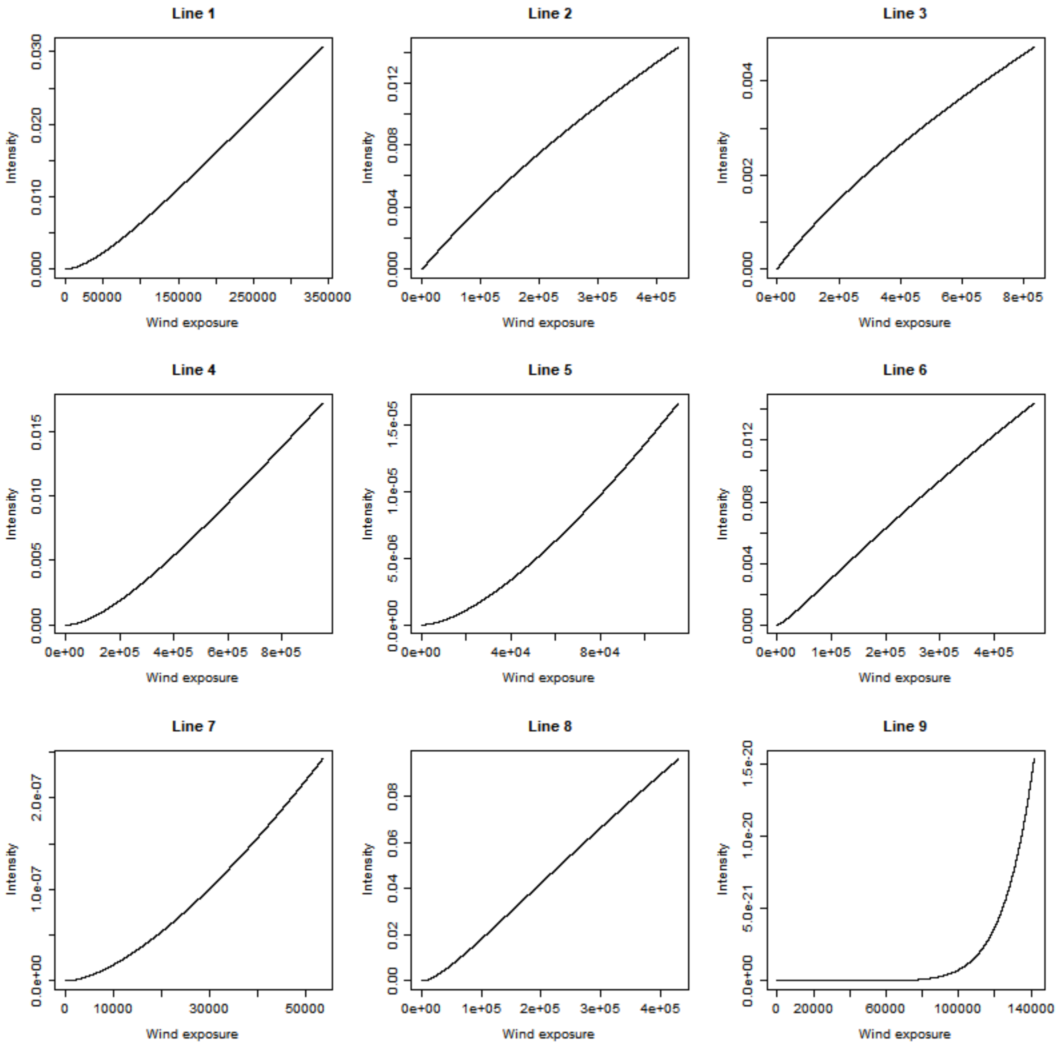

Figure 9: For each line the intensity is plotted as a function of wind exposure. The parameter value of β and the line-specific parameters μ_l and σ_l used to generate these plots are the ones used for the simulated data. Note that the x-axis for a line is only showed for typical values, hence on the interval $[0, \max\, w_{n,l}^t]$, where the max wind exposure is the max value obtained in the period January-June, 2014. Note also that the range of the y-axes differ.

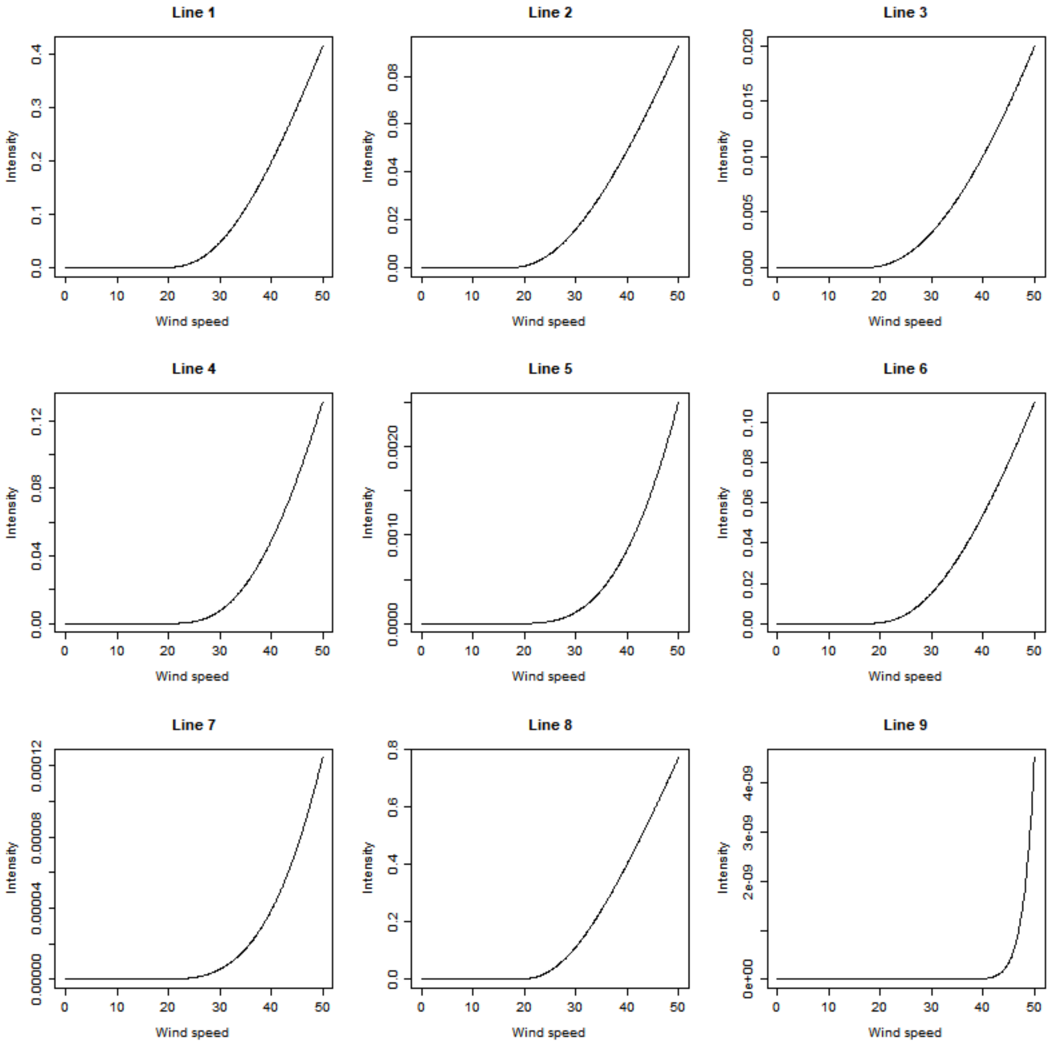

Figure 10: For each line the intensity is plotted as a function of wind speed, here for a given segment length of $d_{n,l} = 150$ m. The parameter value of β and the line-specific parameters μ_l and σ_l used to generate these plots are the ones used for the simulated data. Note that the range of the y-axes differ.

Line	# failures, $u^t_{n,l} > u_{\text{threshold}}$	# failures, $u^t_{n,l} \leq u_{\text{threshold}}$
1	5	0
2	4	3
3	3	1
4	0	0
5	0	0
6	0	0
7	0	0
8	10	0
9	0	0

Table 2: The table gives an overview of the simulated failure data used for the MCMC runs. The failures are here divided into two groups based on wether the corresponding wind speed $u^t_{n,l}$ along the segment that failed for the time of failure was above the wind threshold $u_{\text{threshold}} = 15$ m/s or not.

hour, while for the segment-wise model we need the failures on segment level as input to the likelihood.

5 Algorithm

Having outlined our Bayesian hierarchical models we have obtained expressions for the resulting posterior distributions. These are the distributions of interest as we need them for predictions later on. However, as seen from (4.11) and (4.13) these are a product of many conditional distributions, and by inserting each of the conditional distributions we end up with a non-standard multivariate distribution in each case. To be able to sample from the posteriors we use MCMC as this is a very general approach allowing for sampling from complex distributions. As mentioned in Section 2 there are many choices to be made when implementing an MCMC algorithm. First of all one must decide for the type of updates, in our case chosen as either Gibbs or more general Metropolis-Hastings updates, and then for the type of proposal distributions. In case of Gibbs the proposal distribution is the full conditional distribution and for the MH case we choose a proposal that is either a normal random walk or the type of uniform distribution discussed earlier.

We introduce how we update each parameter type in a top-down approach, starting at the hyperprior level. For all parameters we first found the unnormalized full conditional distribution, and if it turned out to be a standard distribution and simple to sample from, a Gibbs step was chosen accordingly. However this is only the case for two out of the 22 parameters.

5.1 Gibbs updates

Consider the parameter m_μ. Its full conditional is found by noting that it is proportional to the full joint distribution. We only keep the factors in (4.11) where m_μ is included since all other factors are given, and hence only treated as constants in this case. Recall that $f(m_\mu)$ is given by (4.8) and $f(\mu_l|m_\mu, v_\mu)$ as in (2.19). This yields

$$\begin{aligned}
f(m_\mu|\boldsymbol{x}, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, v_\mu, m_\sigma) &\propto f(\boldsymbol{x}, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \\
&\propto f(\boldsymbol{\mu}|m_\mu, v_\mu) \cdot f(m_\mu) \\
&\propto \prod_{l=1}^{L} f(\mu_l|m_\mu, v_\mu) \\
&= \prod_{l=1}^{L} \frac{1}{\sqrt{2\pi v_\mu}} e^{-\frac{(\mu_l - m_\mu)^2}{2v_\mu}} \\
&\propto e^{-\sum_{l=1}^{L} \frac{(\mu_l - m_\mu)^2}{2v_\mu}} \\
&= e^{-\frac{1}{2v_\mu}(\sum_{l=1}^{L} \mu_l^2 - \sum_{l=1}^{L} 2m_\mu\mu_l + \sum_{l=1}^{L} m_\mu^2)} \\
&\propto e^{-\frac{1}{2v_\mu}(-2m_\mu \sum_{l=1}^{L} \mu_l + \sum_{l=1}^{L} m_\mu^2)} \\
&= e^{-\frac{1}{2v_\mu}(Lm_\mu^2 - 2m_\mu \sum_{l=1}^{L} \mu_l)} \\
&= e^{-\frac{1}{2(\frac{v_\mu}{L})}(m_\mu^2 - \frac{2}{L} m_\mu \sum_{l=1}^{L} \mu_l)} \\
&\propto e^{-\frac{1}{2(\frac{v_\mu}{L})}(m_\mu^2 - \frac{2}{L} m_\mu \sum_{l=1}^{L} \mu_l)} \cdot e^{-\frac{1}{2(\frac{v_\mu}{L})}(\frac{1}{L} \sum_{l=1}^{L} \mu_l)^2} \\
&= e^{-\frac{1}{2(\frac{v_\mu}{L})}(m_\mu - \frac{1}{L} \sum_{l=1}^{L} \mu_l)^2}
\end{aligned}$$

where we have explicitly multiplied with an exponential factor on the second to last line above to clearly show that what we end up with must be a normal distribution. We are allowed to do this since this factor is a constant. Hence

$$m_\mu|... \sim \mathrm{N}(\frac{1}{L}\sum_{l=1}^{L} \mu_l, \frac{v_\mu}{L})$$

where $\frac{1}{L}\sum_{l=1}^{L} \mu_l$ is the mean and $\frac{v_\mu}{L}$ the variance.

The full conditional is also easily found for v_μ. We have that $f(v_\mu)$ is given by (4.9) and obtain the following full conditional

$$\begin{aligned}
f(v_\mu|\boldsymbol{x}, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, m_\sigma) &\propto f(\boldsymbol{x}, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \\
&\propto f(\boldsymbol{\mu}|m_\mu, v_\mu) \cdot f(v_\mu) \\
&\propto \frac{1}{v_\mu} \prod_{l=1}^{L} f(\mu_l|m_\mu, v_\mu) \\
&= \frac{1}{v_\mu} \prod_{l=1}^{L} \frac{1}{\sqrt{2\pi v_\mu}} e^{-\frac{(\mu_l - m_\mu)^2}{2v_\mu}}
\end{aligned}$$

$$\propto \frac{1}{v_\mu^{\frac{L}{2}+1}} e^{-\frac{\sum_{l=1}^{L}(\mu_l - m_\mu)^2}{2v_\mu}}$$

which when compared to the probability density function (2.16) is seen to be an inverse-gamma distribution. The corresponding shape parameter is $L/2$ and scale parameter $\left(\sum_{l=1}^{L}(\mu_l - m_\mu)^2\right)/2$, hence

$$v_\mu|... \sim \text{IG}(\frac{L}{2}, \frac{\sum_{l=1}^{L}(\mu_l - m_\mu)^2}{2}).$$

5.2 Metropolis-Hastings updates

For the rest of the parameters we use Metropolis-Hastings updates as the full conditionals are found to be non-standard. As the parameter m_σ must be positive we use the uniform proposal distribution given in (2.12) with corresponding tuning parameter denoted by r_{m_σ}. Note that in the expression for the corresponding acceptance probability (2.13) all factors of the posterior that do not contain m_σ cancel out, and we are left with the factors that matter, hence

$$\begin{aligned} f(m_\sigma|\boldsymbol{x}, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu) &\propto f(\boldsymbol{x}, \alpha, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \\ &\propto f(\boldsymbol{\sigma}|m_\sigma) \cdot f(m_\sigma) \\ &\propto \prod_{l=1}^{L} f(\sigma_l|m_\sigma) \end{aligned}$$

where $f(m_\sigma)$ is given by (4.10) and $f(\sigma_l|m_\sigma)$ by (2.17).

The same type of proposal distribution is chosen for β, and we now denote the tuning parameter by r_β. The corresponding full conditional is proportional to

$$\begin{aligned} &f(\beta|\boldsymbol{x}, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \propto f(\boldsymbol{x}|\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}) \cdot f(\beta) \\ &\propto \begin{cases} \prod_{t=1}^{T}\prod_{l=1}^{L} f(x_l^t|\beta, \mu_l, \sigma_l) \text{ for line-wise model,} \\ \prod_{t=1}^{T}\prod_{l=1}^{L}\prod_{n=1}^{N_l} f(x_{n,l}^t|\beta, \mu_l, \sigma_l) \text{ for segment-wise model.} \end{cases} \end{aligned}$$

The parameter μ_l can be either positive or negative, and we therefore assign it a normal random walk as proposal distribution. Let s_μ be the corresponding standard deviation in the normal random walk, and hence the proposal distribution is given as (2.18) with the mean being the current value of μ_l. Note that the tuning parameter is the same for all μ_l's. The corresponding full conditional is proportional to

$$\begin{aligned} &f(\mu_l|\boldsymbol{x}, \beta, \mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_L, \sigma_1, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \propto f(\boldsymbol{x}|\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}) \cdot f(\mu_l|m_\mu, v_\mu) \\ &\propto \begin{cases} \left[\prod_{t=1}^{T} f(x_l^t|\beta, \mu_l, \sigma_l)\right] \cdot f(\mu_l|m_\mu, v_\mu) \text{ for line-wise model,} \\ \left[\prod_{t=1}^{T}\prod_{n=1}^{N_l} f(x_{n,l}^t|\beta, \mu_l, \sigma_l)\right] \cdot f(\mu_l|m_\mu, v_\mu) \text{ for segment-wise model} \end{cases} \end{aligned} \tag{5.1}$$

where we only include the factors of the likelihood which are dependent on the line-specific parameter μ_l, hence the data for only this line. The acceptance probability is given by (2.11) in this case.

As σ_l is a standard deviation it must necessarily be positive, and hence we choose the uniform proposal distribution (2.12) with corresponding tuning parameter r_σ and acceptance probability (2.13). Note that the tuning parameter is the same for all σ_l's. In the same manner as before we obtain only the factors which remain in its full conditional distribution when omitting all factors that otherwise cancel out. This yields

$$\begin{aligned} &f(\sigma_l|\boldsymbol{x}, \alpha, \beta, \mu_1, ..., \mu_L, \sigma_1, ..., \sigma_{l-1}, \sigma_{l+1}, ..., \sigma_L, m_\mu, v_\mu, m_\sigma) \propto f(\boldsymbol{x}|\beta, \boldsymbol{\mu}, \boldsymbol{\sigma}) \cdot f(\sigma_l|m_\sigma) \\ &\propto \begin{cases} \left[\prod_{t=1}^T f(x_l^t|\beta, \mu_l, \sigma_l)\right] \cdot f(\sigma_l|m_\sigma) \text{ for line-wise model,} \\ \left[\prod_{t=1}^T \prod_{n=1}^{N_l} f(x_{n,l}^t|\beta, \mu_l, \sigma_l)\right] \cdot f(\sigma_l|m_\sigma) \text{ for segment-wise model.} \end{cases} \end{aligned} \tag{5.2}$$

6 Implementation

After one has formulated a model and decided for a type of update for each parameter or block of parameters the next step is implementing the algorithm. We did our implementation of the MCMC algorithm in `C++` since the use of objects suits this kind of implementation and as we are interested in speed. MCMC is in general computational intensive. In addition, the simulated data set of failures was also obtained from code written and run in `C++`. However, `R` is used for most of the plots and visualizations in the result section, in addition for making predictions. For the latter we read text files generated from the MCMC runs in `C++` into `R` and base the predictions on these samples. The initial and necessary preparation of data sets, including wind data, segment lengths and true observed failure data, was done using `Python`. Thereafter the data required for the MCMC runs was written to files on a desired format, which were then again read into `C++`. In this section we start by giving the exact updating scheme used, hence the order and number of updates per iteration for the MCMC runs. Following this we give an overview of the overall structure of our implementation, along with some comments on implementation-specific choices made. Lastly we shortly comment on one way to speed up computations, namely by utilizing the possibility of parallel computation.

6.1 Updating scheme for MCMC

As mentioned earlier there are many possible combinations regarding the order of updates in addition to how one defines which updates and how many of each that are part of what we refer to as one iteration. When we first implemented the MCMC routine we did so in a component-wise manner, updating each parameter once per iteration using the proposal distributions and acceptance probabilities described in the previous section. However, it soon turned out that the updates which require the calculation of the likelihood or some factors of it were computationally hard and time-consuming compared to the rest. This is the case for β, μ_l and σ_l. However, for the parameters at the hyperprior level the updates are much more efficient. It is therefore of interest to avoid getting in a situation where the convergence to the posterior distribution takes a lot of time due to slow exploration of the parameter space of one or several of the hyperparameters. Hence we decide to update these parameters more frequently than the rest, and set to 1000 times more. One iteration is then initially 1000 updates of each hyperparameter and one update of each of the remaining parameters. However, since we plan to run through a huge number of iterations we also want to avoid that too much time is spent on writing parameter values to file and we want to try to keep the corresponding file size on a decent level, because huge files requires more time to be read into R. We therefore perform thinning before running the code, such that only every 10th iteration is written to file. From now on, when we mention iteration we refer to the already thinned chained and hence the generated parameter values that were actually written to file. As a result, the total number of times the parameters have been updated is then way more than what it first seems from trace plots displayed in the result section.

Recall that for the Metropolis-Hastings updates in our algorithm we have a tuning parameter for each type of update. The values for these are chosen to be

$$\begin{aligned} r_{m_\sigma} &= 1.05 \\ r_\beta &= 1.05 \\ s_\mu &= 0.1 \\ r_\sigma &= 1.05 \end{aligned} \tag{6.1}$$

based on the experience of slow exploration along each parameter axes if choosing larger ones. However, their exact values are somewhat arbitrarily chosen, and on purpose rather set smaller than larger to avoid a chain that stays in the same state for many subsequent iterations. Since we have simulated data we know the true values of all parameters at the prior level in our model. Some shorter runs were run with arbitrary initial values, which did not cause any problems. However, for the long run we choose to start at their true values. At least we then expect these initial values to be a good starting point, and the time until convergence should not be slowed down by this choice. Recall that the starting point does not affect convergence itself, however the time until convergence is reached will be different if starting far away from the parameter regions with high posterior densities. However, the hyperparametres are given somewhat arbitrary initial values. Summarized this gives the following scheme, where we let m denote the iterations for the already thinned chain.

1. Set the initial values for all parameters. In our case we let β, μ_l and σ_l start as their true value, hence the ones in (4.14). Otherwise we somewhat arbitrarily set
   $m_\mu = 3.1$
   $v_\sigma = 15.1$
   $m_\sigma = 3.1$

2. For m=1,2,....

   For i=1,...,10

```
(a) For $k$=1,...,1000
        i. Update $m_\mu$
       ii. Update $v_\mu$
      iii. Update $m_\sigma$
(b) Update $\beta$
    For $l$=1,...,L
        i. Update $\mu_l$
       ii. Update $\sigma_l$
```

6.2 A possible way to code the MCMC scheme

The aforementioned updating scheme can be implemented in many ways, and we present here our approach. We implemented the aforementioned updating scheme in `C++`. Each of the parameters are represented by an object, with the most recent parameter value stored as a member variable of the object. When updating a parameter it is then the member variable of the corresponding object that is potentially changed. The value of this member variable is written to file for every iteration m, and hence we only need the most current one to be stored in the object itself. Note that $\mu_1, ..., \mu_l$ are all instances of the same class, and the same goes for $\sigma_1, ..., \sigma_L$. Instances of the same class have the same properties in the sense that they share the same member functions. We let among others the calculation of the prior or hyperprior distribution be a member function, but only the factors of it that are needed, see the previous section. Note that this in practice is implemented as the logarithm of the corresponding function. In addition we create all the constructors such that one sets the initial value when creating an instance from that class. Other member functions that we need are `get` and `set` functions such that it is indeed possible to get, and hence use the value currently stored in the object in calculations, and to be able to update the member variable through the `set` function.

For each update we need to propose a potential new value, to calculate the associated acceptance probability and last but not least update the corresponding parameter object if the proposal is accepted. To organize these updates we again use objects from different classes. Let a proposal class consist of its tuning parameter as a member variable and with member functions `propose`, `accept` and `update`. In addition the calculation of the proposal density is needed for the `accept`-function. The constructor is created such that one needs to assign the corresponding tuning parameter a fixed value when creating an instance. Note that $\mu_1, ..., \mu_l$ are all updated using the same proposal instance, the same goes for $\sigma_1, ..., \sigma_L$. For a Gibbs update the corresponding proposal class does necessarily not have a tuning parameter as member variable. For each update done in the updating scheme the `update`-function of the corresponding proposal object is called with the parameter object as input for which we want an update. Inside the `update`-function a new value is first proposed by calling the `propose`-function, then the probability of accepting this proposal is calculated by the `accept` function, and lastly the member variable of the parameter object is updated if the proposal was accepted. Note that all other objects needed for calculating the acceptance probability must be given as input parameters to `update`.

Since the failure data, along with the wind data and segment length data for calculating wind exposure, only appears in the likelihood, it is natural also to create a class for the likelihood. We let the data be stored as member variables, which are initialized when constructing an instance of it at the very beginning, and these are naturally kept unchanged. The likelihood object is used as input wherever needed. The calculation of the likelihood is naturally one of its member functions. Since the update of μ_l and σ_l only requires some factors of the likelihood as seen in (5.1) and (5.2), we also include the calculation of these factors as an own member function. This is time-saving, as calculating the full likelihood would be unnecessary for these updates.

We let the `update`-function in all proposal classes return either 1 or 0 depending on whether the proposed new value is accepted or not. In this way we keep track of the acceptance rate for each parameter within each iteration m. Hence, for each parameter we sum the total number of proposals accepted within iteration m and divide this sum by the total number of updates for this parameter within the same iteration. We also write the log-posterior probability density for the set of parameters generated for each iteration to file together with the log-likelihood and log-joint prior. This is done to give an overview of whether the chain is moving in the right direction, i.e. towards regions of higher posterior densities. This is at least what one wants to see at the beginning of the chain if starting at values far from the most probable ones. For all draws needed from the uniform and normal distribution we use built-in functions in `C++`. To draw from an inverse-gamma distribution we use the built-in gamma distribution and transform the resulting sample as explained in the theory section to obtain a sample from an inverse-gamma distribution.

For the readers not experienced with implementation of MCMC methods one should note that all densities are in practice computed as the logarithm of the density. This is done to avoid over- and underflow (Gelman et al., 2014). The acceptance probability in the Metropolis-Hastings algorithm is in practice computed as the logarithm of (2.9), hence as $\ln f(\tilde{\theta}|x) - \ln f(\theta|x) + \ln q(\theta|\tilde{\theta}) - \ln q(\tilde{\theta}|\theta)$. Recall that the segment-wise intensities are of the form $-\ln\Big(1 - F(w^t_{n,l}(\beta)|\mu_l, \sigma_l)\Big)$, as seen from the terms in (4.6). From (2.21) we see that these can be expressed as

$$-\ln\left(\Phi\left(\frac{-(\ln w^t_{n,l}(\beta) - \mu_l)}{\sigma_l}\right)\right).$$

However, if one first calculates the cumulative distribution $\mathbf{\Phi(\cdot)}$ and then take the logarithm $\ln \mathbf{\Phi(\cdot)}$ we still loose precision when having potential small or large values for the argument of the cumulative distribution (Linhart, 2008). Hence, we need an approximation for the logarithm of the standard normal distribution, and we choose to use one of the three different implementations described in Linhart (2008), all written in `C`. This algorithm is named `lnnorm` and uses different approximations to the logarithm on different intervals of the real line. Since this is the one out of the proposed functions that works the best for the tails of the normal distribution and since it was relatively straightforward to incorporate in our set-up, we chose this. The code for the corresponding algorithm is available as source material to the aforementioned article.

6.3 Parallel computing

The calculation of the likelihood is computationally heavy, in the sense that it requires iterations over all segments for all lines for all hours. One can compute this sequentially using a standard triple for loop. However, to avoid an unnecessary slow program we choose to parallelize the code. Recall that we assume the number of failures within different hours independent, and thus each hour's contribution to the likelihood can be calculated independently of other hours. Given a line, its contribution to the likelihood is on log-form a sum of hour-wise contributions. The order in which each hour's contribution is calculated has no impact on the sum, and hence we let these be computed in parallel. For this we use OpenMP, which is an application programming interface that enables parallel computations for programs written in `C++`, `C` and `Fortran` (Barbara Chapman and van der Pas, 2008). It assigns different operations to be done on separate processors or cores of processors. Most importantly; it is easy to use and only required little change to our initially sequential-written code.

In our case we let each hour's contribution to the likelihood be stored as an element of a vector. After all parallel computations are done we obtain the likelihood by summing all elements of the vector. Even when the likelihood is computed through parallel computations, the program takes time to run. The running time is of course dependent upon how many CPUs one has available. We ran our code using a calculation server available for math students, and since the resources must be shared and vary with the use of it by others the number of CPUs available was not constant at all times.

7 Results

We run one long MCMC run for both the line-wise and segment-wise model. Recall that the same prior and hyperprior distributions are used for both models, see Section 4. In addition, the corresponding types of proposal distributions are chosen to be the same and as given in Section 5, with exact same values for the tuning parameters, see (6.1). The order of updates follows the updating scheme outlined in the previous section. The output is in each case a chain of parameter values that is converging in distribution towards the target distribution, hence the posterior. One should run for several initial values as discussed before to be more confident when assessing convergence. Due to limited time and computational resources we do so for some shorter runs. These runs indicate that convergence is obtained after an acceptable number of iterations. Then we run one longer run for each model. We base all the following results and discussions on these long runs.

In this section we analyze the results from the MCMC runs for both models. First we assess convergence and discuss our choice of burn-in period based on MCMC diagnostic plots. In addition we comment on the chain's mixing properties. To summarize the posterior distribution the marginal posterior distribution for each parameter is visualized as a histogram of the generated values and the corresponding 90% credible interval is given. Following this, we proceed with predictions based on the output from the MCMC runs. The probability of at least one failure within a future hour is naturally one of the desired quantities to predict. We evaluate the predictions in terms of the logarithm of the absolute error and compare both models. Lastly we discuss how the whole procedure of MCMC runs and predictions could be done in practice. This includes using the newest information available and finding a way to evaluate predictions that works when not having any true reference values, as is the case when having simulated data.

7.1 Posterior distribution

The MCMC updating scheme was on purpose set to run for a very large number of iterations such that we could terminate the computation when desired. Let the total number of iterations be denoted M when not counting the initial value. After our run this is equal to $M_{\text{LINE}} = 193199$ for the line-wise model and $M_{\text{SEGMENT}} = 52229$ for our segment-wise model. These numbers differ substantially mainly because their corresponding computations have had a somewhat different amount of CPUs available.

7.1.1 MCMC diagnostics and mixing properties

To determine a reasonable burn-in period we look at trace plots. As stressed in the theory section, this must be done by evaluating all parameters, and hence we investigate the trace plot for each parameter of a model. These plots are found in the Appendix as there are 22 parameters in total for each model. We start by looking at the trace plots for the MCMC run based on the segment-wise model, displayed in Figures 19 to 40. To easier assess the burn-in period we plot the trace plot for the first 5000, 10000 and 20000 iterations in addition to for the full simulation. Several of these plots indicate that the burn-in must be at least 4000-5000 iterations long, see for instance the plots for μ_1, σ_1, μ_8 and σ_8 in Figures 23, 24, 37, and 38, respectively. However, from the full trace plots of m_μ and v_μ in Figures 19 and 20 a larger burn-in period seems appropriate as the first 10 000 iterations or so appear to be distinguishable in their look from the remaining iterations.

All in all this suggests a burn-in period of about 10 000 iterations. Since 10 000 is already almost a fifth of all iterations we find this large enough. Note that even though some of the trace plots at first glance seem to have a "periodic" trend in the way they are fluctuating, this is not the case as these fluctuations happen at random and not as a multiple of some integer. In addition we have no strong trends. Some of the marginal posterior distributions show sign of having a somewhat heavy tail as some peaks in the trace plot are seen getting higher and higher as the number of iterations increases, see the plots for parameter v_μ and β in Figures 20 and 22.

The running mean plots show that at least for some parameters, see for instance v_μ, μ_4 and μ_6 in Figures 20, 29 and 31, the running mean has not yet stabilized. Recall from the definition of the running mean that we take the mean of all values up to and including an iteration. Thus, it is naturally sensitive to the burn-in period and any major fluctuations early on as the total number of values at those times is relatively small. However, if one had run the chain for much longer we expect that these means would stabilize as then the burn-in would influence the mean less. There is no evident indication of non-convergence from the trace plots, and hence we treat the chain has having obtained convergence from after the burn-in period.

In the same manner we investigate the trace plots corresponding to the parameters of the line-wise model. These are plotted for the first 5000, 10000 and 50000 iterations in addition to for the full simulation and are found in Figures 41 to 62. There is in this case no clear indication of exactly where to set the end of the burn-in period. From most of the trace plots the parameter values generated seem to quickly reach a zone within they vary, however for parameter σ_3 and σ_5 in Figures 50 and 54 it seems reasonable to maybe discard the first 10 000 iterations. For the line-wise model we have many more iterations than for the segment-wise one and do have many generated values left even if discarding several thousands. Unlike the segment-wise model, the running mean plots for the line-wise model do seem to stabilize. This is not surprising, as the total number of iterations in this case is almost four times as large. Hence, we conclude that the running mean plots do indeed support the overall indication of convergence.

As computation time is often of interest we would like to obtain a representative sample from the posterior distribution in an efficient way. Hence it is not enough that the distribution of the chain has reached convergence, one also wants to run long enough to explore the whole support of the posterior distribution. If this is done in an efficient manner, then we say that the chain mixes well. Recall that we keep track of the acceptance rate for each parameter within each outer loop $m.$, i.e. the proportion of accepted proposals among the total proposals within that iteration. To find the overall acceptance rate we simply discard the samples belonging to the burn-in period and take the mean of all the remaining iteration's acceptance rates to find the overall acceptance rate. Recall also that m_μ and v_μ have acceptance rates equal to 1 due to Gibbs updates. For the rest of the parameters the overall acceptance rates lie in the range of 0.64-0.98 for the parameters in the line-wise model and 0.63-0.98 for the segment-wise model. Most rates are definitively above the recommended values and twelve of the parameters have in both cases corresponding acceptance rates above 90%. For each parameter the corresponding acceptance rate is similar for both models which might be due to the fact that the models are very similarly formulated and since we also have chosen the exact same tuning parameter values for both runs. That the acceptance rates tend to be relatively large indicate that the majority of the proposals are accepted. Hence, the chain explores the parameter space in a relatively slow manner compared to what it could have done. We did not perform any extensive tuning of the tuning parameters of the Metropolis-Hastings proposal distributions since this would require many additional shorter runs.

If a chain is mixing well one should see that the autocorrelations go towards zero as the lag increases (Johnson and Albert, 1999). Since the acceptance rates are relatively high in our case we would expect the autocorrelations to go faster towards zero if choosing a somewhat larger value for the tuning parameter in the proposal distributions. From most of the autocorrelation plots, see for instance Figures 22, 39, 44 and 61, it appears as if there is some underlying periodicity, but we know by construction of the Markov chain that this is not a property of the chain. Hence, this must be due to a random effect.

7.2 Marginal posterior distributions and cross-correlations

We let as mentioned the burn-in period be $B = 10000$ for both models such that the iterations $\theta^{(0)}, \theta^{(1)}, ..., \theta^{(B)}$ are discarded. The remaining parameters are then treated as coming from the posterior distribution. This is in our case a multivariate distribution of 22 parameters. However, to be able to summarize it we first look at the marginal posterior distributions. This is the distribution of one parameter given the data, hence a univariate distribution. Recall that for a given parameter θ_i the generated values $\theta_i^{(B+1)}, ..., \theta_i^{(M)}$ do necessarily come from the marginal posterior, hence from $f(\theta_i|x)$. In the Appendix in Figure 63-68 the histograms from the chain of each parameter are displayed. The histograms based on the segment-wise model are plotted directly beneath the ones for the line-wise one to make it easier to compare them. We see that the shape of the histograms are similar for most parameters, the ones for the segment-wise model appear to be in general a bit wider, and often somewhat shifted towards slightly larger values. Note particularly that for β the histogram corresponding to the line-wise model in Figure 63 has a much heavier tail than for the segment-wise one.

To better summarize the marginal posterior distributions we give their corresponding 90% equal-tail credible intervals in Table 3. The built-in `quantile`-function in `R` is used for this as the 90% credible interval consists of the 5%- and 95%-quantiles. These intervals confirm what we have already seen from the histograms of the marginal posterior samples in Figure 63-68. For 19 out of the 22 parameters the intervals for the segment-wise model are shifted towards higher values in addition to being wider. The β-parameter stands out as having very different intervals for the two models, the corresponding 90% credible interval for the line-wise model is 4.7 times as wide as the interval for the segment-wise model. For the latter it is in contrast to most of the other intervals shifted towards smaller values. All in all the segment-wise model manages to restrict the range of β the most, while in general the line-wise model is the one with least uncertainty about the rest of the parameters, except for σ_8. Note that since the prior distributions chosen are the exact same for both models, in addition to the simulated data, what makes these credible intervals differ from model to model is only due to the difference in likelihood. Recall that the lines which fail in our six months period of simulated data are lines 1, 2, 3 and 8, see Table 2. Keeping this in mind when having another look at Table 3 we also clearly see that when only considering the line-specific parameters μ_l and σ_l the intervals corresponding to the parameters for the lines that do fail is narrower than for the rest. This is the case for both models.

From the credible intervals one sees within which range a parameter value typically lie. How they are spread is seen better from the histogram of the generated values which resembles the corresponding marginal posterior density. Another aspect to look at is correlations. In Figures 11 and 12 the cross-correlation plot for each pair (μ_l, σ_l) is shown for the line-wise and segment-wise model, respectively. These are the parameters in the

	Posterior quantiles		
	5%	95%	CI width
m_μ	14.83	20.42	5.58
v_μ	0.51	29.81	29.30
m_σ	0.69	2.73	2.04
β	3.45	47.30	43.85
μ_1	13.48	16.63	3.15
σ_1	0.64	1.91	1.27
μ_2	15.02	19.18	4.16
σ_2	1.34	2.74	1.40
μ_3	15.93	19.79	3.86
σ_3	1.37	2.68	1.31
μ_4	14.63	23.27	8.64
σ_4	0.09	1.82	1.72
μ_5	14.00	23.56	9.56
σ_5	0.09	1.76	1.67
μ_6	14.28	25.50	11.21
σ_6	0.12	1.63	1.50
μ_7	14.27	24.27	9.99
σ_7	0.23	2.26	2.03
μ_8	13.26	15.59	2.33
σ_8	0.60	1.72	1.12
μ_9	14.02	25.46	11.44
σ_9	0.05	1.88	1.83

(a) Line-wise model

	Posterior quantiles		
	5%	95%	CI width
m_μ	16.07	22.97	6.90
v_μ	0.04	41.19	41.15
m_σ	0.99	3.69	2.70
β	0.75	10.08	9.34
μ_1	14.18	17.96	3.77
σ_1	1.01	2.42	1.41
μ_2	16.47	21.90	5.42
σ_2	2.01	3.63	1.63
μ_3	17.03	22.12	5.10
σ_3	1.89	3.47	1.58
μ_4	15.40	26.45	11.05
σ_4	0.14	2.24	2.10
μ_5	15.09	26.59	11.50
σ_5	0.12	2.36	2.23
μ_6	15.07	29.06	14.00
σ_6	0.17	2.15	1.98
μ_7	15.42	26.52	11.10
σ_7	0.31	2.88	2.57
μ_8	14.59	17.40	2.81
σ_8	1.38	2.44	1.06
μ_9	15.06	28.42	13.36
σ_9	0.07	2.48	2.41

(b) Segment-wise model

Table 3: 90% equal-tail credible intervals for all parameters in a) the line-wise model and b) the segment-wise model. These intervals are found based on quantiles of the generated values from the MCMC runs for both models.

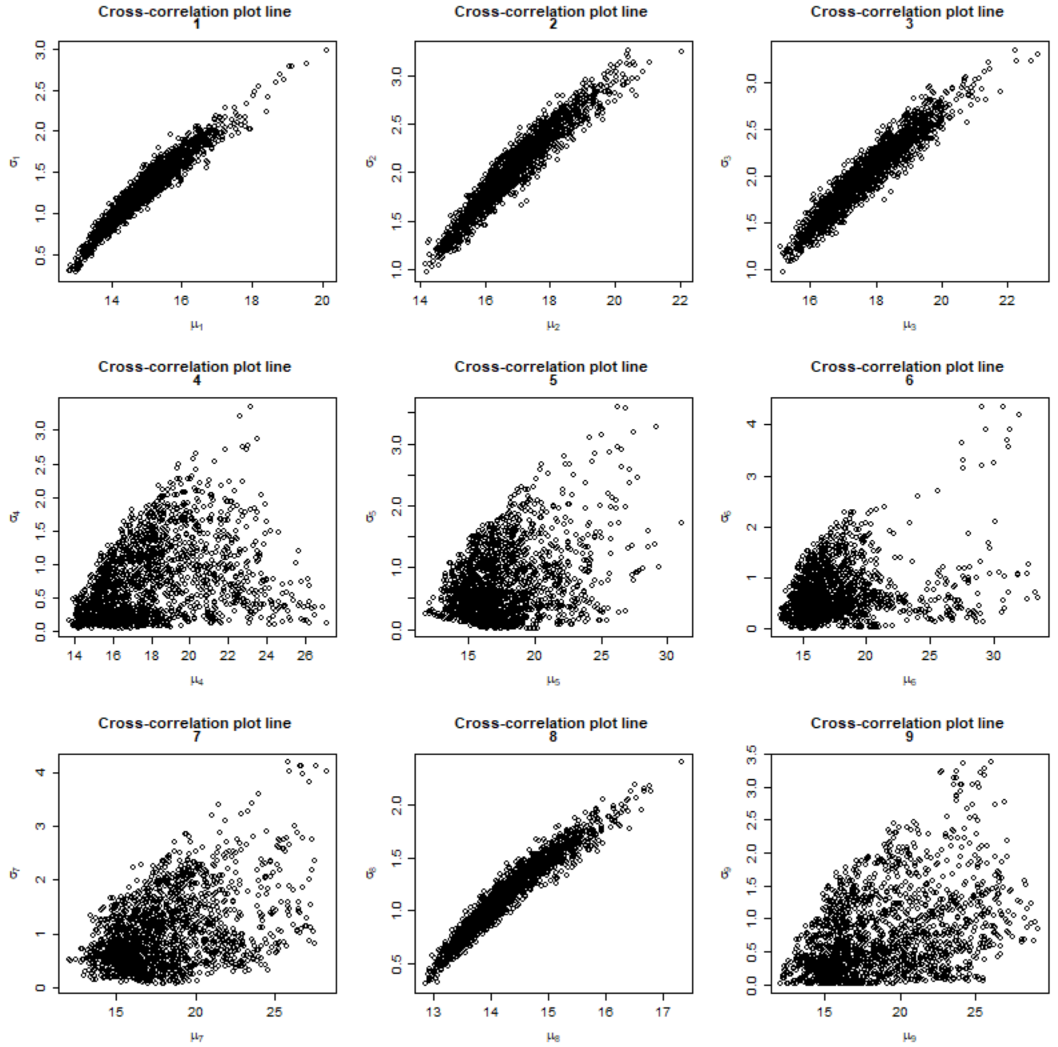

Figure 11: Cross-correlation plots of the pair (μ_l, σ_l) for all lines based on the MCMC run for the line-wise model. To avoid too many points in the plot we have only extracted every 100th iteration from the MCMC run after first having discarded the burn-in period.

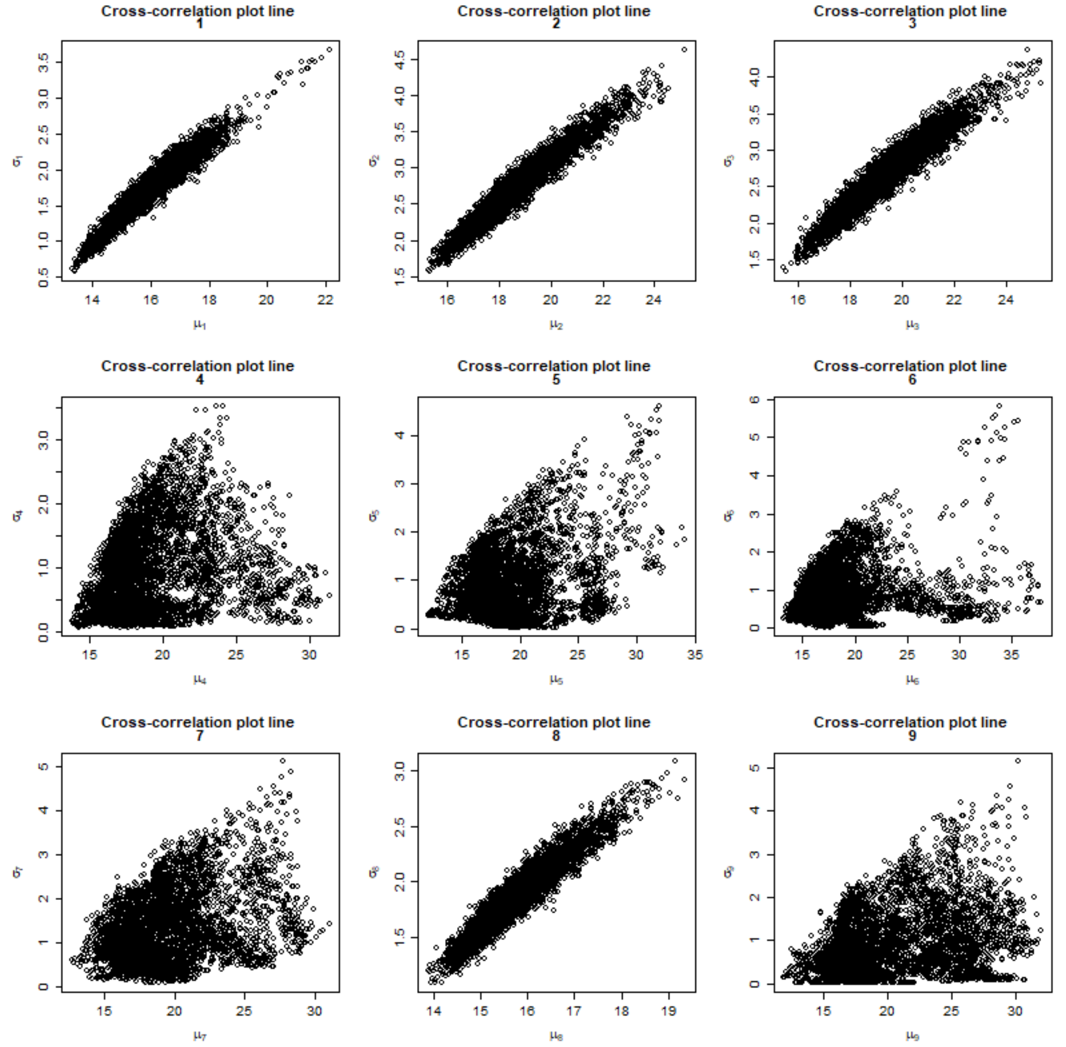

Figure 12: Cross-correlation plots of the pair (μ_l, σ_l) for all lines based on the MCMC run for the segment-wise model. To avoid too many points in the plot we have only extracted every 100th iteration from the MCMC run after first having discarded the burn-in period.

cumulative lognormal function, which again are connected to the probability of failure in our models. For the cross-correlation plots only every 100th iteration for the line-wise model and every 10th iteration for the segment-wise model were plotted. Otherwise it would have been difficult to distinguish areas with many versus few generated values. In the posterior histograms in Figure 63-68 one can observe the tails of the marginal distributions. However, what is not observable here is the tails of the joint distribution of μ_l and σ_l. These can be seen from Figures 11 and 12 as the areas in which there are the fewest points.

It is clear that for the lines for which we have failures, i.e. lines 1, 2, 3 and 8, there is a positive correlation between the line-specific parameters. In those cases for a given value of μ_l the corresponding σ_l only varies within a seemingly small interval, and vice versa. The correlation is present in the plots for the other lines as well, but not as strong. We observe that for all lines there are no points in the upper left area, i.e. σ_l is never relatively large for the smaller values of μ_l. For the lines without failures there are however still many points in the lower right part of the plots. Recall that in the formulation of our model the parameters are assumed conditionally independent given their parent nodes. Hence for μ_l and σ_l the corresponding joint prior is

$$f(\mu_l, \sigma_l | m_\mu, v_\mu, m_\sigma) = f(\mu_l | m_\mu, v_\mu) f(\sigma_l | m_\mu, v_\mu)$$

which is a product of each of their prior distributions. Since we have assumed independence between these parameters given the hyperparameters we have not incorporated any correlation between them in our model. Despite this assumption the cross-correlation plots based on the generated values from the posterior joint distribution clearly indicate that these parameters are correlated. Hence, it is the data through the likelihood that must be the reason for this outcome. As this correlation seem to be a common feature for all lines, and particularly the lines that fail, one should take this into account if ever formulating a new model.

In a potential new model one could have modified the prior distributions for μ_l and σ_l to rather be a prior for the pair of (μ_l, σ_l). Then correlation can be incorporated between the parameters within a pair while still letting each pair be conditionally independent of all other pairs. In that case the prior had been a bivariate distribution in contrast to the univariate priors defined in our case. The correlation parameter would then be common for all pairs. Introducing correlation into our already constructed framework would increase the overall amount of parameters in our model. However, the clear trends from the cross-correlation plots indicate that this might still be a reasonable choice. Note also that independence between the parameters within a pair is a special case and would be included in the potential new model, so we would then not discard the possibility of independence.

Recall that we plotted the wind exposure against intensity in Figure 9 for the parameter values chosen for the simulated data set. In the same manner we plot the wind exposure against intensity for typical values of β, μ_l and σ_l based on the MCMC run for the line-wise model. These plots are seen in Figure 13 when the mean is chosen to represent a typical value. Here the range for which we plot is again $[0, \max w_{n,l}]$, but with the maximum wind exposure for a line found by using the mean value of β from the MCMC

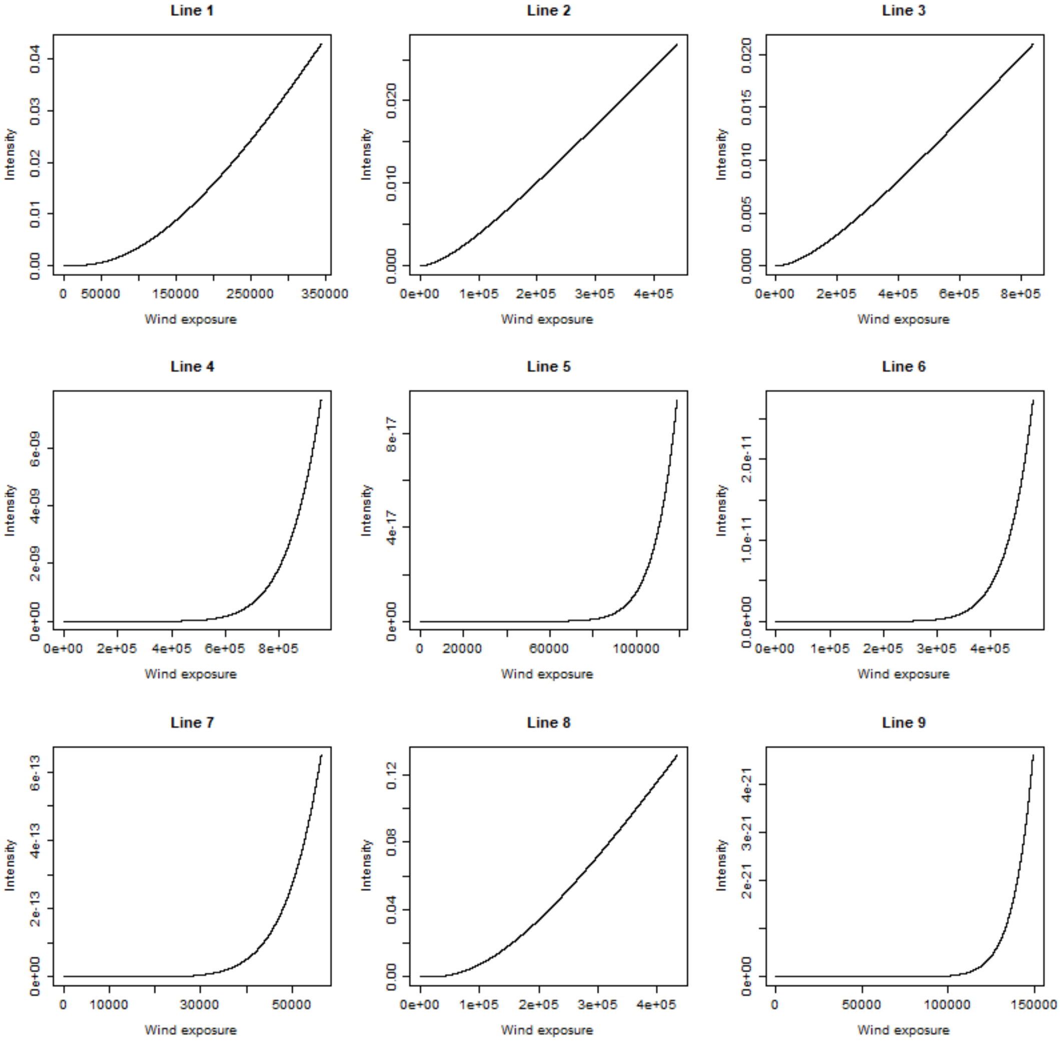

Figure 13: Intensity as a function of wind exposure for all lines based on the line-wise model. The curves are based on the mean values of β, μ_l and σ_l from the corresponding MCMC run. The range of wind exposure is here slightly extended compared to for the case when $\beta = 1$, as is the case for the simulated data.

run. This value of β is then used for calculating all wind exposures for the six months period based on segment lengths and wind data available, and by letting max $w_{n,l}$ be the maximum of those values. The corresponding plot for the segment-wise model is found in the Appendix in Figure 69. What is evident from these plots is that the shape of a line's curve is clearly different for the lines that failed compared to the curves for the lines that did not fail. For lines 1, 2, 3 and 8 there is a steady increase in the intensity, while for the rest of the lines the intensity stays rather small and constant before it increases for the largest wind exposures. However, by noting that the y-axes differ the increase is then not even close to the general increase for the lines that fail. Most interesting is it rather to compare each line's intensity curve to the ones in Figure 9. In that figure all curves, expect for line 9, have a similar shape. Note how the range of the y-axis for a plot for a given line differ in Figure 9 versus the corresponding plot in Figure 13, such that for the lines that fail we have an increase in the max intensity in the latter case. This increase corresponds to for instance a factor 1.4 for line 1 and 1.9 for line 2. For the lines that do not fail we observe that except for line 9 the max intensity is rather lowered, and by a factor of for instance $1.9 \cdot 10^{-9}$ for line 6 and $2.6 \cdot 10^{-6}$ for line 7.

To just look at the mean values from the MCMC runs only give one plausible intensity curve per line. To gain more insight in how much the shape of the intensity curves can in fact vary due to the spread in the generated values from the posterior distribution we also plot several intensity curves in the same plot. As the wind exposure is dependent on β we rather plot these curves as a function of wind speed, and for the line-wise model the corresponding plots are seen in Figure 14. Here the segment length is again set to $d_{n,l} = 150$ m, as in Figure 10. Only the intensity curves corresponding to β, μ_l and σ_l for every 400th iteration of the MCMC run are included. The same type of plots based on the MCMC run for the segment-wise model are found in Figure 70. Note that the y-axes in these plots differ, so one can not directly compare the plots for different lines. The thicker black curve is the same as in Figure 10, hence the one corresponding to the parameter values of the simulated data set. We see that these curves indeed lie inside the range of the possible curves that the parameters from the posterior distribution can take. However, for some lines, see lines 2 and 3, the black curve does not lie in the most probable area, hence not where there are the most curves. From Figure 14 one clearly sees a difference in when the intensities really start to increase. For lines 1, 2, 3 and 8 all lines seem to increase for a wind speed about 20-25 m/s, while for the other lines this range is much wider. Overall, the range of different shapes of the curves is relatively wide. If one had run MCMC for the same models with more data one would believe this range to be less wide.

To be able to compare lines to each other we also plot the same plots for common axes, see Figure 15. From these plots we see that the possible intensity curves for the different lines somewhat overlap each other. This means that even for a line without failures, the same intensity curve as for a line that does fail is probable. However, only for the lines that do not fail can the intensity be very low even for high wind speeds. Keep in mind that when no failures are observed, the contribution to the likelihood for both models for each hour is

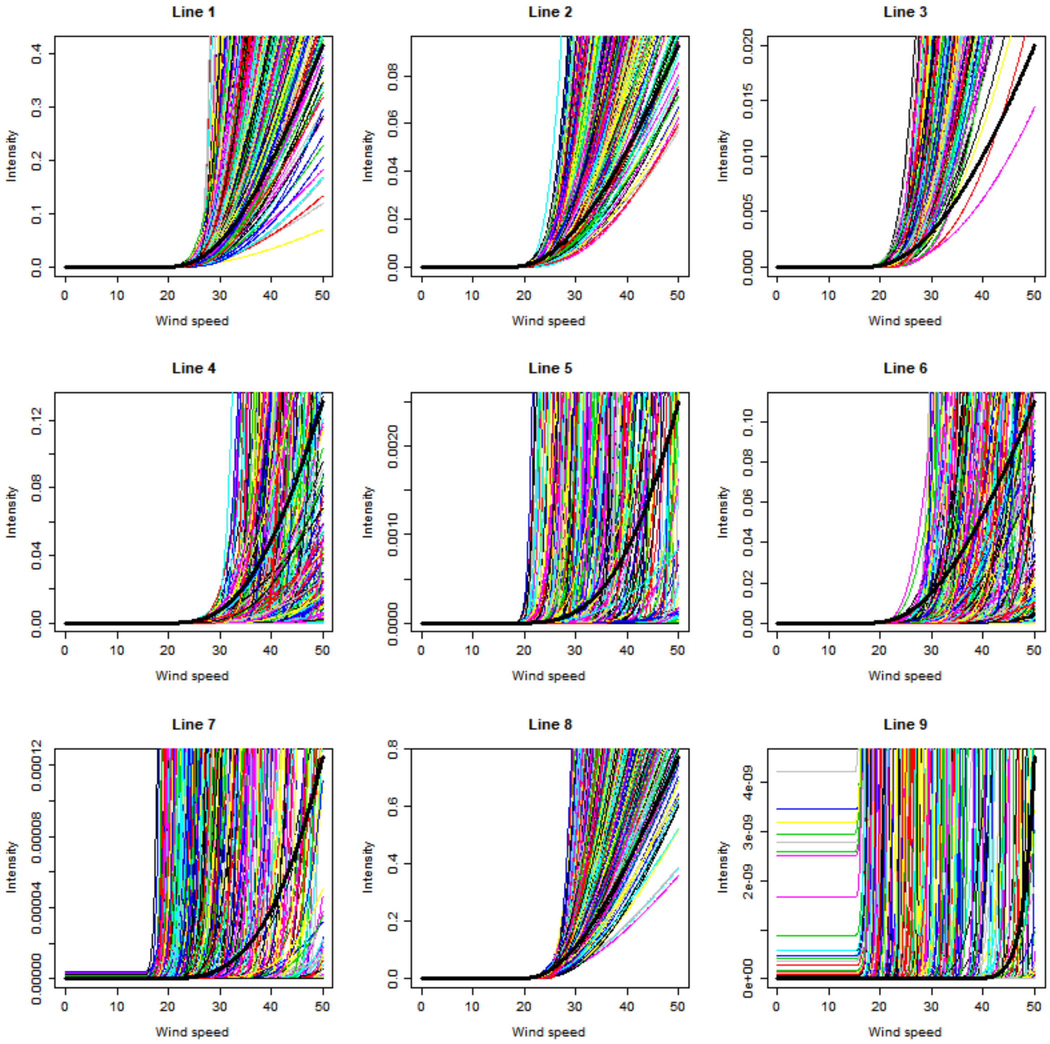

Figure 14: The intensity curve corresponding to every 400th iteration of the MCMC run for the line-wise model. The thicker black line corresponds to the curve the parameters for the simulated data would give. Note that the y-axis differ from plot to plot.

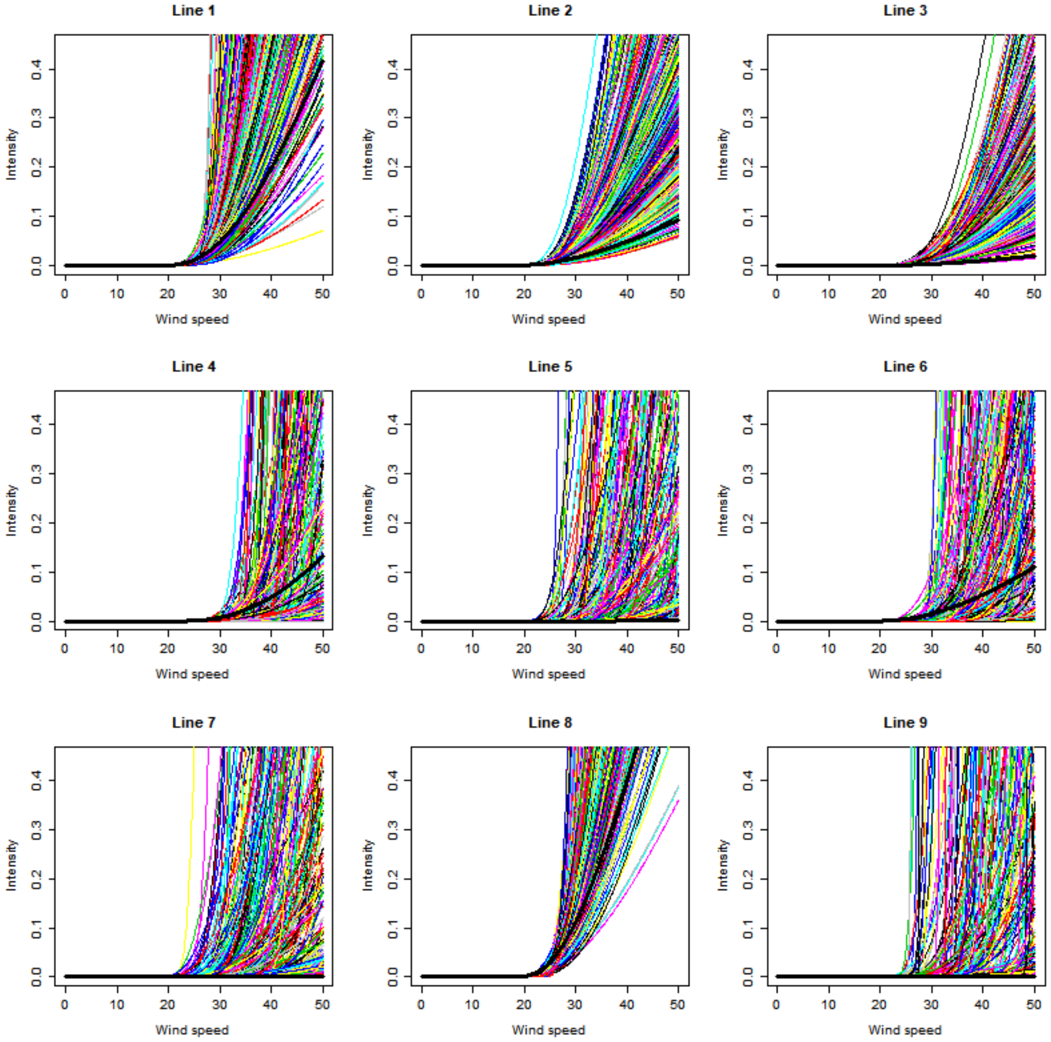

Figure 15: The intensity curve corresponding to every 400th iteration of the MCMC run for the line-wise model. The thicker black line corresponds to the curve the parameters for the simulated data would give. Here the y-axes are set equal for all plots to make it easier to compare different lines.

$$e^{-\lambda_l^t} = e^{-\sum_{n=1}^{N_l} \lambda_{n,l}^t},$$

which of course is large when all segment-wise intensities $\lambda_{n,l}^t$ are small.

7.3 Predictions

We proceed by using the parameter values generated from the aforementioned MCMC runs for predictions. As mentioned earlier we are interested in a way to predict the probability of failure for future times. First we present how this is done and how one can quantify the uncertainty in the prediction for a given hour. Recall that since we have simulated data we know the true parameters, and hence use this to find the error between the prediction made and the true probability of at least one failure. To summarize all predictions and to be able to compare the predictions based on our two models we compute the logarithm of the absolute error for each prediction. We initially believe there to be more information in the segment-wise model as one in the line-wise one "overlook" the information of exactly where the failures occurred.

Note that we predict mostly for illustrative reasons in this report as we have used simulated failure data and a given wind data set. In practice one would run the MCMC algorithm based on true observed failure data. In addition the wind data needed would then rather be weather forecasts, hence there would be some uncertainty in these weather predictions as well. Recall however that we treat wind speed as given in our models and do not incorporate any extra uncertainties due to the weather forecast themselves.

7.3.1 Predict the probability of failure for future hours

We predict the probability of failure on an hourly basis. Recall that for a given hour t the probability of at least one failure along that line is given by (4.7). Based on the parameters generated from the MCMC runs, i.e. $(\beta^{(B+1)}, {\mu_1}^{(B+1)}, ..., {\mu_9}^{(B+1)}, {\sigma_1}^{(B+1)}, ..., {\sigma_9}^{(B+1)}), ..., (\beta^{(M)}, {\mu_1}^{(M)}, ..., {\mu_9}^{(M)}, {\sigma_1}^{(M)}, ..., {\sigma_9}^{(M)})$ we get one corresponding probability of failure for each set of parameters. We denote these by ${p_l^t}^{(B+1)}, ..., {p_l^t}^{(M)}$ for a given line l. The corresponding 90% equal-tail credible interval for p_l^t is then again found from the 5%- and 95%-quantiles. This interval reflects the uncertainty of p_l^t.

Based on the given wind data set we predict the probability of at least one failure on an hourly basis for three periods of three consecutive days. These days are July 1-3, 2014, December 1-3, 2014, and February 10-12, 2015. In Figures 16 and 17 we display the histograms of these values for the first eight hours of February 10, 2015, for line 1. Both the histogram corresponding to the line-wise and segment-wise model are plotted for each hour. The corresponding 90% equal-tail credible intervals are given in Table 4. We see that the magnitude of the predictions can vary from hour to hour, see for instance the predictions for the first two hours, p_1^1 and p_1^2. The shape of the histograms differ from the line-wise model to the segment-wise model in the sense that in the latter case the the most probable predictions are not necessarily as close to zero as for the line-wise model. From the credible intervals all intervals except for the first hour are the widest

	Posterior quantiles		
Hour	5%	95%	CI width
1	6.00E-15	3.06E-04	3.06E-04
2	4.69E-07	1.32E-02	1.32E-02
3	2.84E-04	9.04E-02	9.01E-02
4	6.85E-05	3.72E-02	3.72E-02
5	6.16E-04	5.90E-02	5.84E-02
6	1.43E-04	3.53E-02	3.51E-02
7	4.27E-07	9.06E-03	9.06E-03
8	1.86E-07	7.97E-03	7.97E-03

(a) Line-wise model

	Posterior quantiles		
Hour	5%	95%	CI width
1	6.77E-13	2.64E-04	2.64E-04
2	1.42E-04	3.09E-02	3.08E-02
3	5.90E-03	1.54E-01	1.48E-01
4	1.73E-03	7.08E-02	6.91E-02
5	6.28E-03	9.37E-02	8.74E-02
6	2.75E-03	5.99E-02	5.72E-02
7	1.08E-04	2.12E-02	2.11E-02
8	6.14E-05	1.95E-02	1.95E-02

(b) Segment-wise model

Table 4: 90% equal-tail credible intervals for the predictions made based on a) the line-wise model and b) the segment-wise model for the probability of at least one failures for the first eight hours of February 10, 2015.

for the segment-wise model. However, here we only looked at eight hours for one line, and hence this is nothing to draw any general conclusions from. For predictions for 72 hours we could plot 72 such histograms per model, and find their corresponding 90% equal-tail credible intervals. However, as we have nine lines in total we do not include all of this. See the Appendix for the eight first hours of July 1, 2014, and December 1, 2014, in Figure 71. Their corresponding credible intervals are shown in Table 6.

We have predicted the probability of at least one failure, but one could predict other quantities if that is of interest. For instance the probability of three or more failures within an hour or maybe just the failure intensities. The procedure is similar, one make use of the generated parameter values from the MCMC run and compute the quantity of interest for each set of parameters.

7.3.2 Evaluate predictions

In practice one is probable most interested in one value to represent the probability of failure. This also makes it easier to summarize predictions and quantify how well the models do. As an estimate of the probability of at least one failure for line l in hour t,

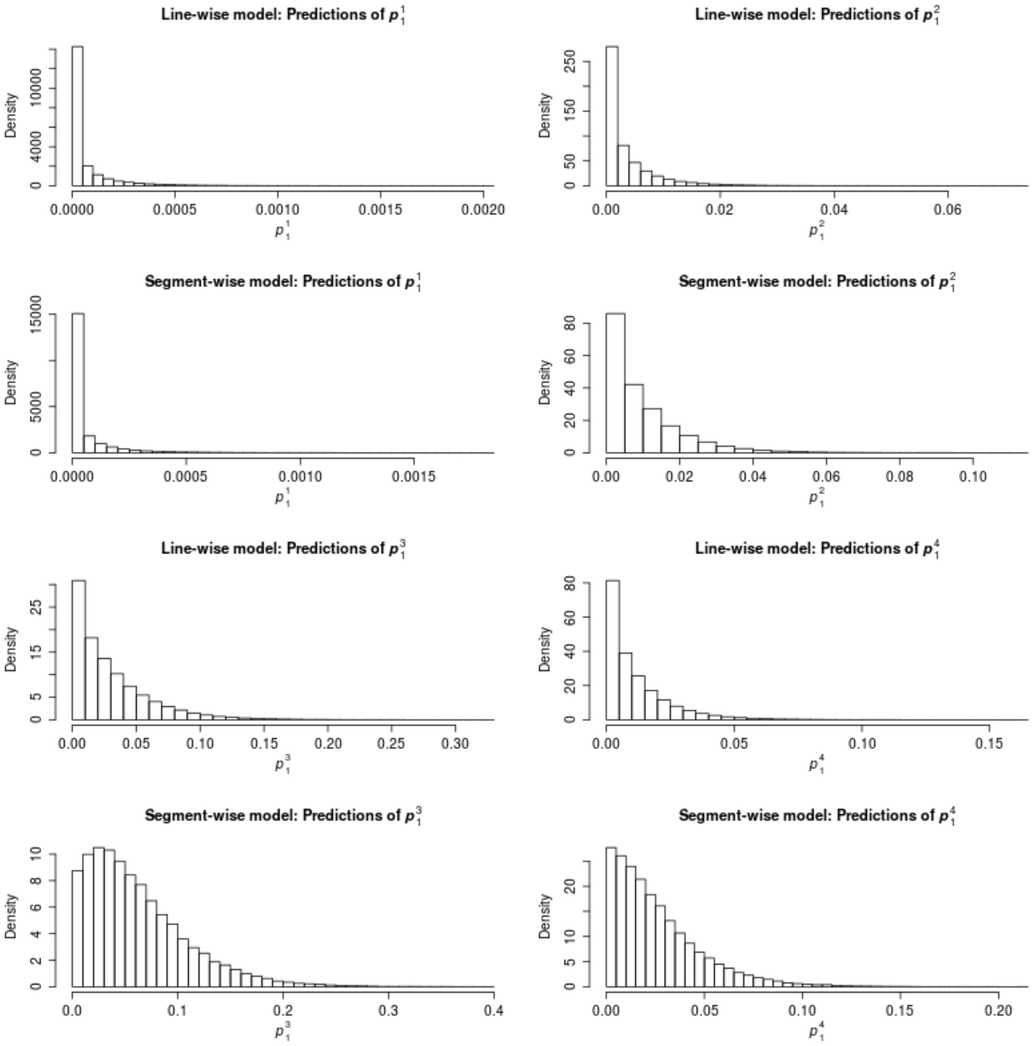

Figure 16: Histograms of the predictions for the probability of at least one failure for line 1 for the first four hours of February 10, 2015. Both the predictions based on the line-wise and segment-wise model are displayed.

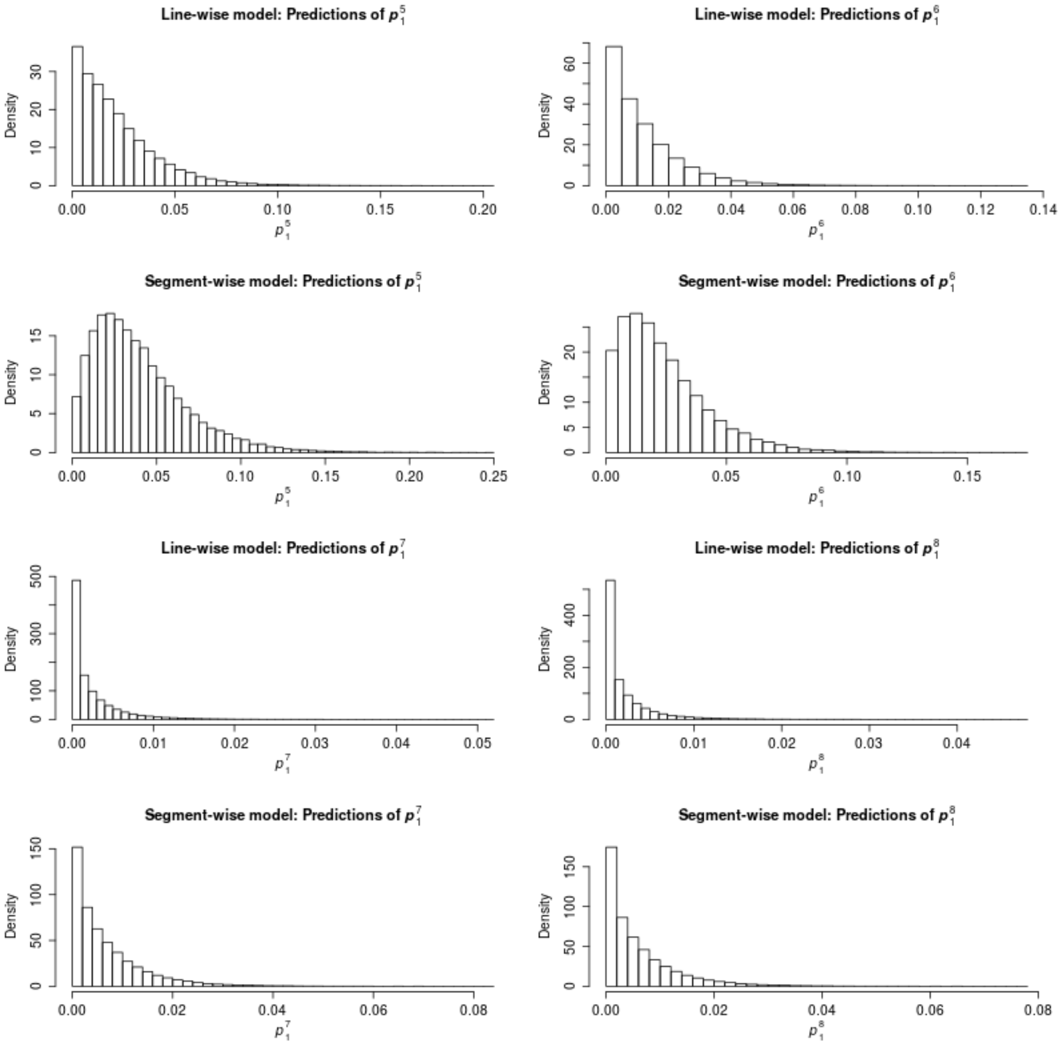

Figure 17: Histograms of the predictions for the probability of at least one failure for line 1 for the fifth to eight first hours of February 10, 2015. Both the predictions based on the line-wise and segment-wise model are displayed.

denoted $\hat{p}_l^t$, we use the mean. Thus, we have that

$$\hat{p}_l^t = \frac{1}{M-(B+1)} \sum_{m=B+1}^{M} p_l^{t(m)} \tag{7.1}$$

which is based on $M-(B+1)$ values computed from the set of parameters from the generated Markov chain. Note that one can use other estimates as well, for instance the median or mode.

We have so far formulated two models and outlined how one based on the MCMC output predicts probability of failures. The next step is to compare the models in terms of how accurate predictions they give. Recall that when we simulated the failure data used for the MCMC runs we let the parameters have the values as shown in (4.14). Based on these we calculate the corresponding probability of failure from (4.7), denoted p_l^t, for each hour we predict for. We treat these as the true probabilities of failure. Hence, it is natural to evaluate how far off the estimated probabilities of failure given in (7.1) are compared to the true ones.

It turns out that the order of magnitude of the true probability and the estimated one for a given hour can differ substantially. This can be seen from Table 5 where we have displayed the values corresponding to the probability of at least one failure within the eight first hours of February 10, 2015. The same is the case for probabilities between different hours as already seen from the histograms in Figures 16 and 17. We therefore choose to look at the logarithm of the absolute error to evaluate the predictions made to avoid that only the larger probabilities dominate. Hence we calculate

$$\ln|\hat{p}_l^t - p_l^t| \tag{7.2}$$

for each hour we predict for. We did so for the 216 hours we predicted for in total, for all lines and for both the line-wise model and the segment-wise model. The logarithm of the absolute errors are summarized in a plot, see Figure 18. Here all $216 \cdot 9 = 1944$ predictions are included. The order of the predictions along the x-axis is so that the predictions for line 1 come first, then line 2 and so on, always with July, December, February as the order of the periods we predict for. We see that there are clearly many hours for which the logarithm of the corresponding absolute error is the same, seen as the horizontal lines in the plot. This happens for all nine lines and is a consequence of the fact that when the wind is below the wind threshold for all line segments, the wind exposure is not a function of wind speed for any of the segments. And hence, since the estimates of the probability of failure are based on the same set of parameters from the MCMC runs for different times, these are going to be equal for a line for all times when the wind is below threshold along the whole line. It is no clear difference between the predictions made from the different models based on the look of this scatter plot as they both have very similar spread in values. Thus, our initial belief that the segment-wise model should yield better predictions is not supported by Figure 18.

One could have thought of different score functions as an alternative way of evaluating the predictions. One then could get one number to summarize the predictions made.

t	p_l^t	$\hat{p}_l^t$	$\ln\lvert\hat{p}_1^t - p_l^t\rvert$
1	6.05E-06	6.21E-05	-9.79
2	1.69E-02	3.39E-03	-4.30
3	1.01E-01	2.98E-02	-2.64
4	4.37E-02	1.14E-02	-3.43
5	6.15E-02	2.15E-02	-3.22
6	3.83E-02	1.20E-02	-3.63
7	1.13E-02	2.35E-03	-4.71
8	1.01E-02	2.01E-03	-4.81

(a) Line-wise model

t	p_l^t	$\hat{p}_l^t$	$\ln\lvert\hat{p}_1^t - p_l^t\rvert$
1	6.05E-06	5.20E-05	-9.99
2	1.69E-02	9.79E-03	-4.94
3	1.01E-01	6.18E-02	-3.23
4	4.37E-02	2.61E-02	-4.04
5	6.15E-02	3.98E-02	-3.83
6	3.83E-02	2.41E-02	-4.25
7	1.13E-02	6.68E-03	-5.37
8	1.01E-02	5.95E-03	-5.48

(b) Segment-wise model

Table 5: The true and estimated probability of at least one failure for the first eight hours of February 10, 2015. Also the logarithm of the absolute error for each prediction is given in the rightmost column.

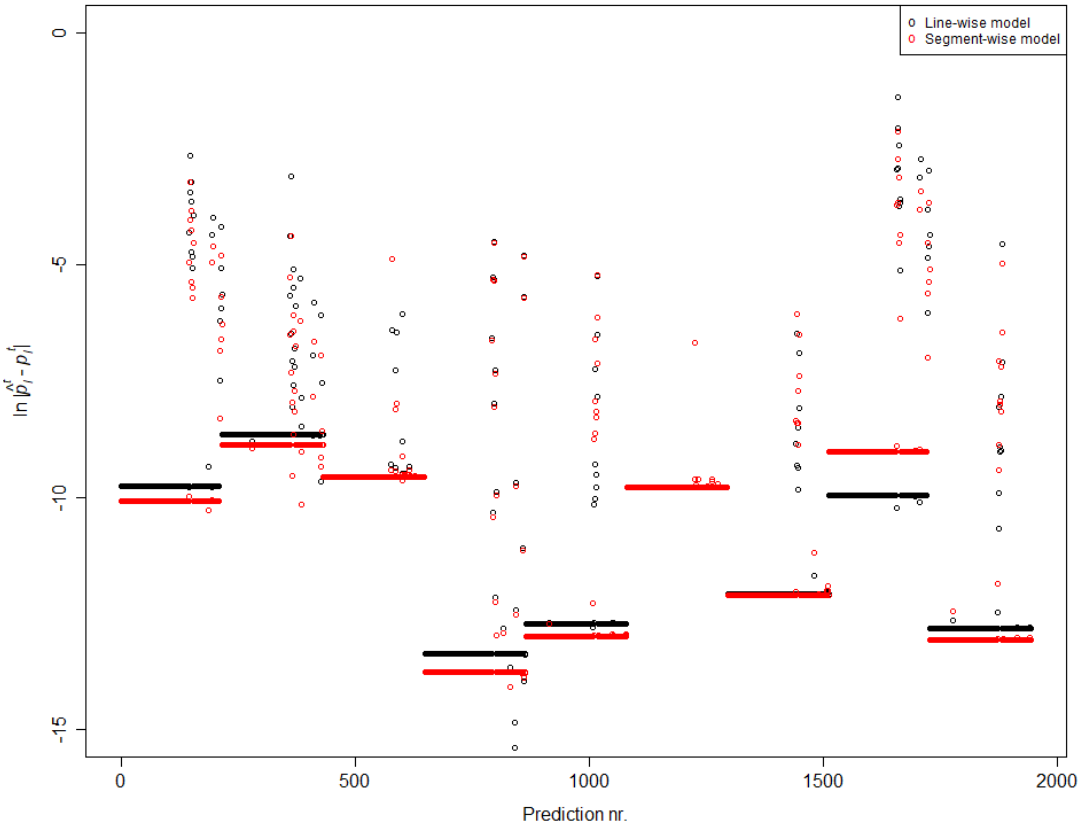

Figure 18: Plot of the logarithm of the absolute error of the predictions made based on the line-wise model in black and the segment-wise model in red. From left to right first all predictions for line 1, then for line 2 and so in is plotted, hence the longer intervals with same value for the logarithm of the absolute error is for the same line. This happens when the wind is below the wind threshold along the whole line for several consecutive hours.

However, if doing so one must keep in mind that the choice of score function affect how one weights different errors. For instance for the square of the error, $(\hat{p}_l^t - p_l^t)^2$, the contributions from the largest probabilities are going to affect the score the most when summing over all predictions made, even though the relative error might be larger for some smaller predictions. Since we do not want to take a stand on which type of predictions should be punished the most, we avoid such score functions at this point.

7.3.3 Update model and score predictions in practice

In practice failures are observed continuously and this makes it possible to update the model at certain intervals to incorporate new information. An update of the model is done according to the model updating scheme in (2.4). One then needs to rerun the MCMC algorithm with an updated likelihood and with the new and previously collected failure data as input. In addition this requires the corresponding wind data, hence one also needs to store weather data for the periods one have observed failures in. Note that even if one had wanted to update the model at a very high frequency, say every three days or every week, there is always a certain delay due to the computation time of the MCMC run. Hence, if this takes several weeks to run then there are naturally observations for that period of time that is not going to be included in the results of the MCMC run. However, if already running the model for many years of data, some weeks or months of data not included is probably not going to make too much difference in the results anyway. How often one should update the model is then a cost-benefit question, and also a question how often failure data is corrected and finally set.

The evaluation of predictions based on our two models in terms of the absolute error (7.2) is only possible in this particular setting, hence when having simulated failure data. In practice we have no true parameters of β, μ_l and σ_l and then this type of evaluation metric makes no sense. However, what does make sense is to compare the predictions made with what one later observes to occurre. An often used score function for predictions is the Brier score (Bradley et al., 2008). If interested in the probability of at least one failure this yields a binary forecast, hence we predict the probability of at least one failure as $\hat{p}_l^t$ and the probability of no failures is then $1-\hat{p}_l^t$. The corresponding Brier score is then given as

$$\text{Brier score } = \frac{1}{N}\sum_{n=1}^{N}(\hat{p}_l^t - I_{x_l^t>0}(x_l^t))^2$$

where the indicator function here assures that the last term evaluates to one if there has been observed more than one failure, hence when $x_l^t > 0$, and to zero otherwise. The lower the Brier score, the better.

8 Closing remarks

In this thesis we have formulated two Bayesian hierarchical models for the number of temporary failures due to wind on overhead transmission lines. They are based on and are an alternative approach to the model in Solheim et al. (2016), which is the one used today by the Norwegian Transmission System Operator. Our models are hierarchical in the sense that they consist of random variables on three different layers and Bayesian since we consider the parameters of the models to be random variables. Wind speed is assumed to be the main explanatory variable and we condition on it, hence we treat it is as known and given. A data set of wind speed is provided by Statnett for nine overhead transmission lines, in addition to the length of each line segment. The number of failures for a line segment within an hour is assumed to follow a Poisson distribution. The corresponding failure rate is connected to wind speed in the way that one sets the probability of at least one failure equal to a function of some parameters and wind exposure. As in Solheim et al. (2016) we let wind exposure be a function of wind speed and segment length. The wind exposure is used as input to the cumulative lognormal function to connect the parameters to the probability of failure. Failures along a segment are assumed independent of failures along other segments and independent of failures within other hours. For the hyperparameters, i.e. the top level of our hierarchy of variables, we assign noninformative and improper priors, and assume all to be independent of each other. The only difference in our two models is that for the segment-wise model the data is on segment-level, hence it requires knowledge of where along a line a failure occurred. The other is a line-wise model where the number of failures is aggregated up to only give the total number of failures for a line within an hour. The latter suits the type of failure data collected by Statnett so far, while the former is an alternative if obtaining more detailed reports on failures in the future.

We combine the likelihood, which represents the distribution of the observable variables, with the prior distribution, i.e. the distribution of the parameters, to find the expression of the corresponding posterior distribution. The posterior distribution we obtain is non-standard and complex, and we find the corresponding integrals needed for finding credible intervals or mean of functions of the parameters analytically intractable. Instead we choose to generate samples from it. For this we choose a Markov chain Monte Carlo approach. By construction, the Metropolis-Hastings algorithm and its special case the Gibbs sampler creates a Markov chain with limiting distribution equal to the posterior distribution. The idea is that initially all parameters are set to an arbitrary value, but by constantly updating them in accordance with an MCMC updating scheme the chain converges in distribution to the posterior. We discard the first iterations, known as the burn-in period, and use the rest as a sample from the posterior.

We run the MCMC method for simulated failure data for a period of six months. The marginal posterior distributions show that there is more certainty about the parameters in the line-wise model than the segment-wise one. For both models we see that the parameters for the lines that did fail in our simulated data set have shorter credible intervals. In addition, there is seen to be a clear positive correlation for each pair of the line-specific parameters that decide the shape of the cumulative lognormal distribution. This is especially the case for the lines that did fail. Further, we predicted the probability of at least one failure for three days in three different months, for all lines. Since we have a set of true parameter values from the simulated data we use these to calculate the corresponding true probabilities of failure. The logarithm of the absolute error for all predictions were displayed in a plot, but there seem to be no clear difference between our two models.

For future work a natural next step is first to investigate the possibility of incorporating a correlation parameter for the pair of the line-specific parameters based on the revelations from the cross-correlation plots in this report. This would require a new prior, a bivariate prior distribution on the prior level in our models. Then the MCMC update scheme could be run on true observed failure data, and preferable for a long period of time. As the failure data is in practice reported on line-level, and since our comparison indicated that there seem to be little difference between our models when it comes to predictions, it is natural to continue with the line-wise model. Since we used simulated data for our MCMC runs, and where the number of failures was similar to the total number of failures for the observed failure data available from Statnett, we do not know how the results would turn out based on true data. It is easy to include more lines than the nine we have been looking at by introducing a new set of line-specific parameters for each line and including all lines in the likelihood. However, MCMC is computational intensive and requires computing power. As one includes more lines and more data, the computations naturally are more time-consuming. In that case one must pay attention to potential ways to optimize the code, and the use of parallel computations. In addition one should compare the model proposed here to the one already in use to see if the predictions made are any better.

References

Barbara Chapman, G. J. and van der Pas, R. (2008), *Using OpenMP: Portable Shared Memory Parallel Programming*, The MIT Press, Cambridge, Massachusetts.

Bolstad, W. M. (2007), *Introduction to Bayesian Statistics*, 2nd edn., John Wiley & Sons, Inc., Hoboken, New Jersey.

Bradley, A. A., Schwartz, S. S., and Hashino, T. (2008), "Sampling Uncertainty and Confidence Intervals for the Brier Score and Brier Skill Score", *Weather and Forecasting*, **23**, 992–1006.

Brier, G. W. (1950), "Verification of Forecasts Expressed in Terms of Probability", *Monthly Weather Review*, **78**, 1–3.

Casella, G. and Berger, R. L. (2002), *Statistical Inference*, 2nd edn., Brooks/Cole, Cengage Learning, Belmont, California.

Chib, S. and Greenberg, E. (1995), "Understanding the Metropolis-Hastings Algorithm", *The American Statistician*, **49**, 327–335.

Chib, S. and Greenberg, E. (1996), "Markov Chain Monte Carlo Simulation Methods in Econometrics", *Econometric Theory*, **12**, 409–431.

Gamerman, D. and Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, vol. 68 of *Texts in Statistical Science Series*, 2nd edn., Chapman & Hall/CRC, Boca Raton, Florida.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014), *Bayesian Data Analysis*, vol. 106 of *Chapman & Hall/CRC Texts in Statistical Science*, 3rd edn., CRC Press, Boca Raton, Florida.

Givens, G. H. and Hoeting, J. A. (2013), *Computational Statistics*, Wiley Series in Computational Statistics, 2nd edn., John Wiley & Sons, Inc., Hoboken, New Jersey.

Hoff, P. D. (2009), *A First Course in Bayesian Statistical Methods*, Springer Texts in Statistics, Springer, New York.

Johnson, V. E. and Albert, J. H. (1999), *Review of Bayesian Computation*, Springer, New York.

Kroese, D. P. and Chan, J. C. (2014), *Statistical Modeling and Computation*, Springer, New York.

Kruschke, J. K. (2015), *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*, 2nd edn., Academic Press, London.

Larsen, R. J. and Marx, M. L. (2018), *An Introduction to Mathematical Statistics and Its Applications*, 6th edn., Pearson Education, Inc., Boston.

Lee, P. M. (2012), *Bayesian Statistics: An Introduction*, 4th edn., John Wiley & Sons, Ltd., Chichester, West Sussex.

Lesaffre, E. and Lawson, A. B. (2012), *Bayesian Biostatistics*, John Wiley & Sons, Ltd., Chichester, West Sussex.

Linhart, J. M. (2008), “Algorithm 885: Computing the Logarithm of the Normal Distribution”, *ACM Transactions on Mathematical Software (TOMS)*, **35**, 20:1–20:10.

Ntzoufras, I. (2009), *Bayesian Modeling Using WinBUGS*, Wiley Series in Computational Statistics, John Wiley & Sons, Inc., Hoboken, New Jersey.

Rubinstein, R. Y. and Kroese, D. P. (2008), *Simulation and the Monte Carlo Method*, Wiley Series in Probability and Statistics, 2nd edn., John Wiley & Sons, Inc., Hoboken, New Jersey.

Solheim, Ø. R., Kjølle, G., and Trötscher, T. (2016), “Wind dependent failure rates for overhead transmission lines using reanalysis data and a Bayesian updating scheme”, 2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS).

Solheim, Ø. R. and Trötscher, T. (2018), “Modelling transmission line failures due to lightning using reanalysis data and a Bayesian updating scheme”, 2018 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS).

Norwegian Ministry of Petroleum and Energy (2019a), ”The electricity grid”, `https://energifaktanorge.no/en/norsk-energiforsyning/kraftnett/`, Online; accessed 26-February-2019.

Norwegian Ministry of Petroleum and Energy (2019b), ”Security of electricity supply”, `https://energifaktanorge.no/en/norsk-energiforsyning/forsyningssikkerhet/`, Online; accessed February 26, 2019.

Norwegian water resources and energy directorate (2016), ”System operation in the Norwegian power system”, `https://www.nve.no/energy-market-and-regulation/system-operation-in-the-norwegian-power-system/`, Online; accessed February 26, 2019.

Statnett SF, seksjon Feilanalyse (2015), ”Årsstatistikk 2014: Driftsforstyrrelser og feil i 33-420 kV-nettet”, `https://www.statnett.no/contentassets/5fb5605039314f498ed16f8561695a0c/feilanalyse-arsstatistikk-2014-33-420-kv.pdf`, Online; accessed February 26, 2019.

Tierney, L. (1994), “Markov Chains for Exploring Posterior Distributions”, *The Annals of Statistics*, **22**, 1701–1728.

Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K. (2012), *Probability & Statistics for Engineers & Scientists*, 9th edn., Pearson Education. Inc., Boston.

Ward, D. M. (2013), “The effect of weather on grid systems and the reliability of electricity supply”, *Climatic Change*, **121**, 103–113.

Appendix

	Posterior quantiles		
Hour	5%	95%	CI width
1	6.00E-15	2.95E-04	2.95E-04
2	6.00E-15	2.95E-04	2.95E-04
3	6.00E-15	2.95E-04	2.95E-04
4	6.00E-15	2.95E-04	2.95E-04
5	6.00E-15	2.95E-04	2.95E-04
6	6.00E-15	2.95E-04	2.95E-04
7	6.00E-15	2.95E-04	2.95E-04
8	6.00E-15	2.95E-04	2.95E-04

(a) Line-wise model

	Posterior quantiles		
Hour	5%	95%	CI width
1	1.27E-13	2.28E-04	2.28E-04
2	1.27E-13	2.28E-04	2.28E-04
3	1.27E-13	2.28E-04	2.28E-04
4	1.27E-13	2.28E-04	2.28E-04
5	1.27E-13	2.28E-04	2.28E-04
6	1.27E-13	2.28E-04	2.28E-04
7	1.27E-13	2.28E-04	2.28E-04
8	1.27E-13	2.28E-04	2.28E-04

(b) Segment-wise model

Table 6: 90% equal-tail credible intervals for the predictions made based on a) the line-wise model and b) the segment-wise model for the probability of at least one failures for the first eight hours of July 1, 2014. The table is the exact same for the first eight hours of December 1, 2014.

Segment-wise model

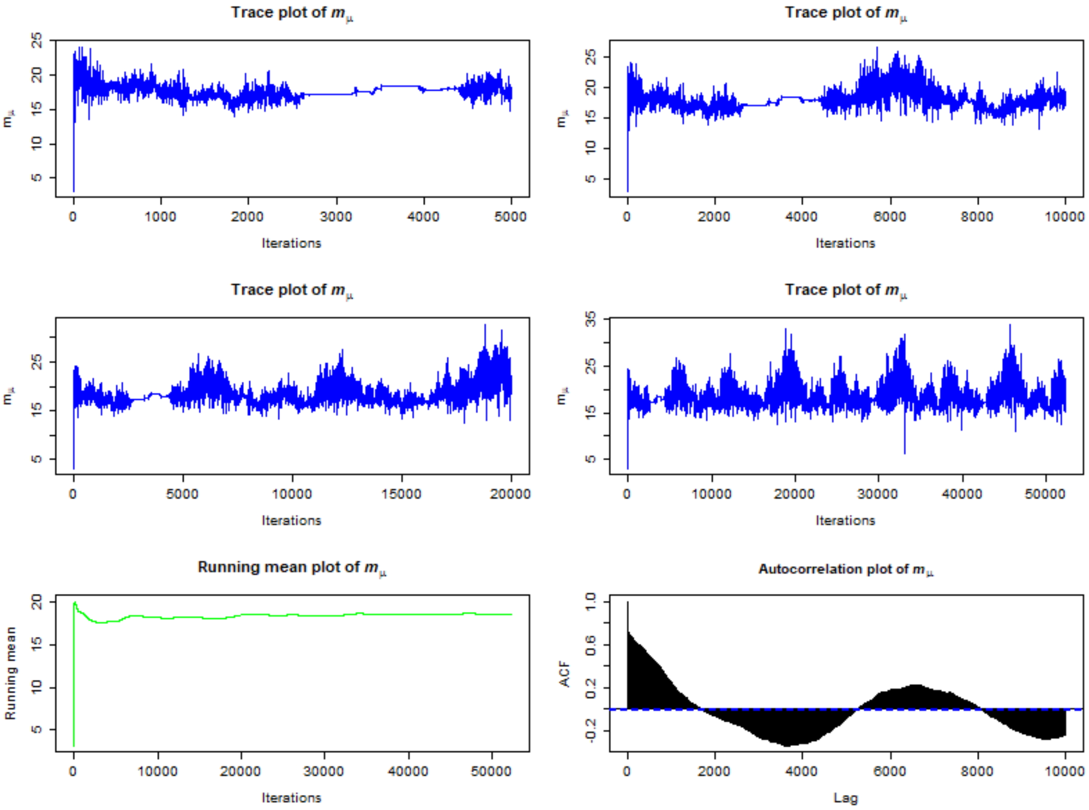

Figure 19: Plots for the parameter m_μ based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

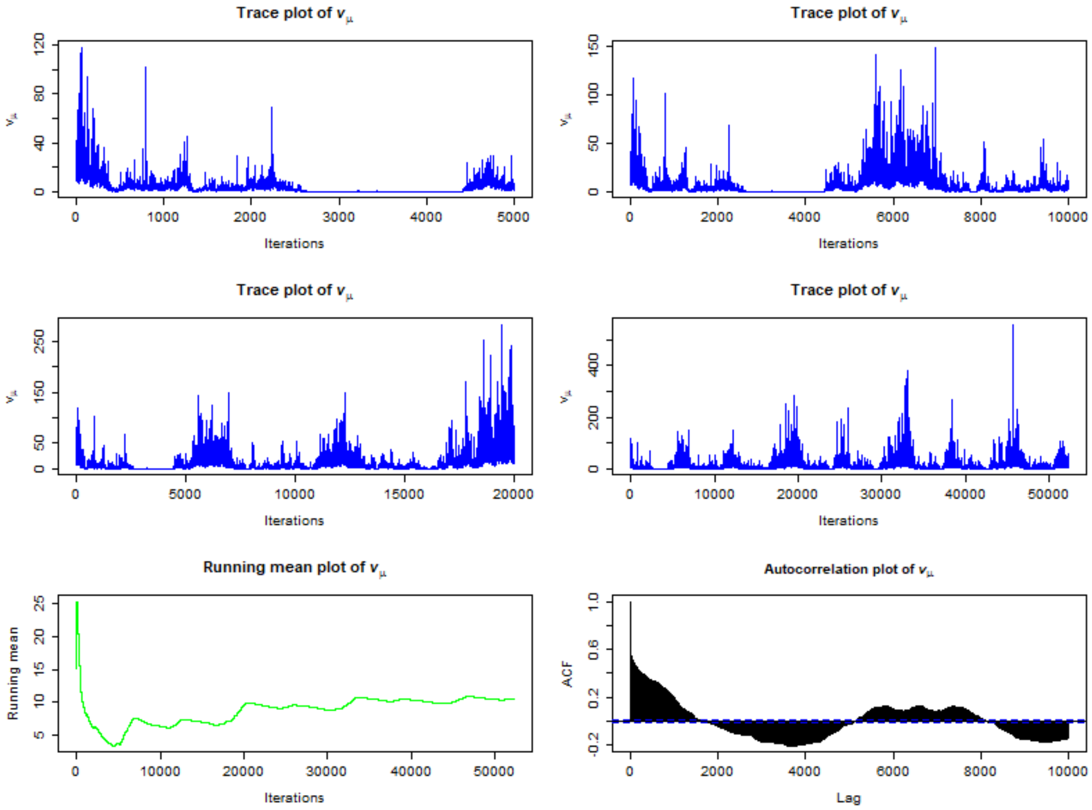

Figure 20: Plots for the parameter v_μ based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

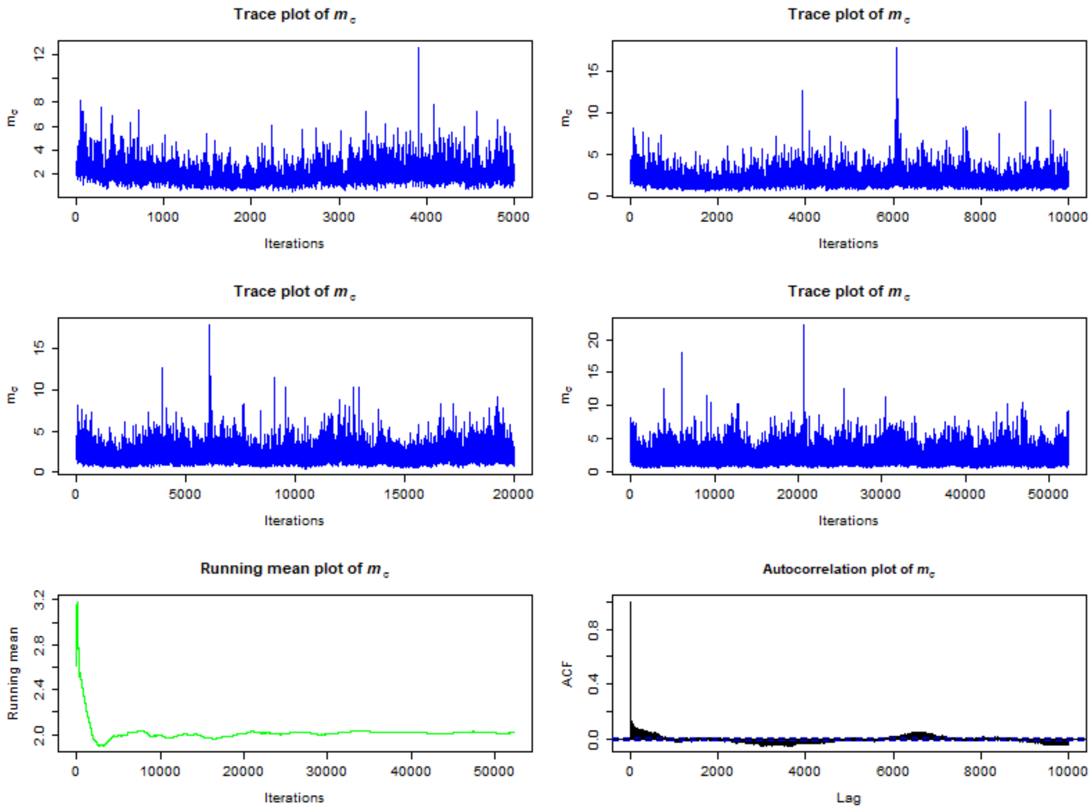

Figure 21: Plots for the parameter m_σ based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

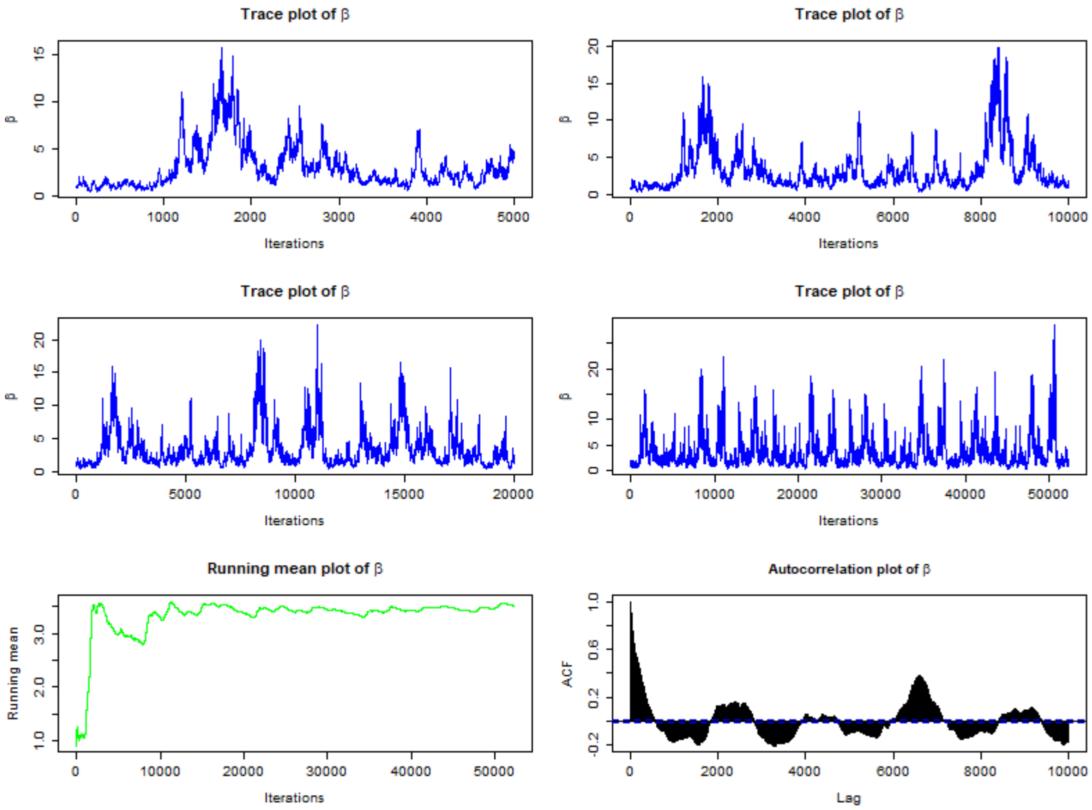

Figure 22: Plots for the parameter β based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

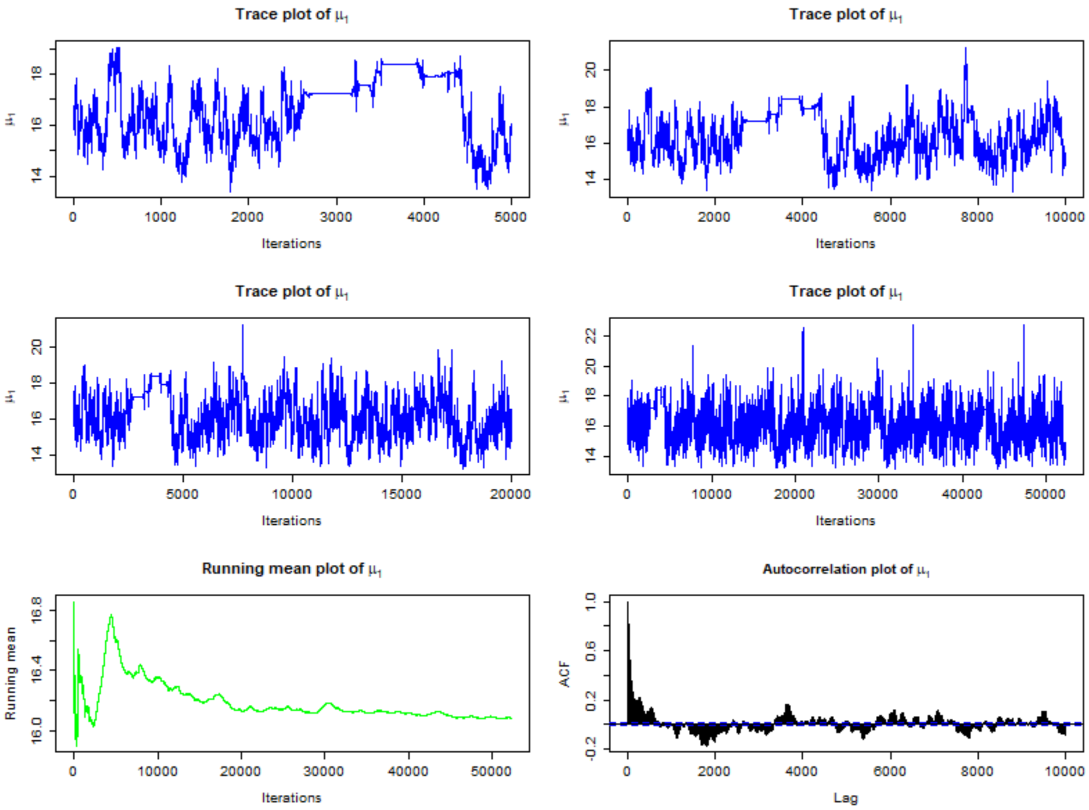

Figure 23: Plots for the parameter μ_1 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

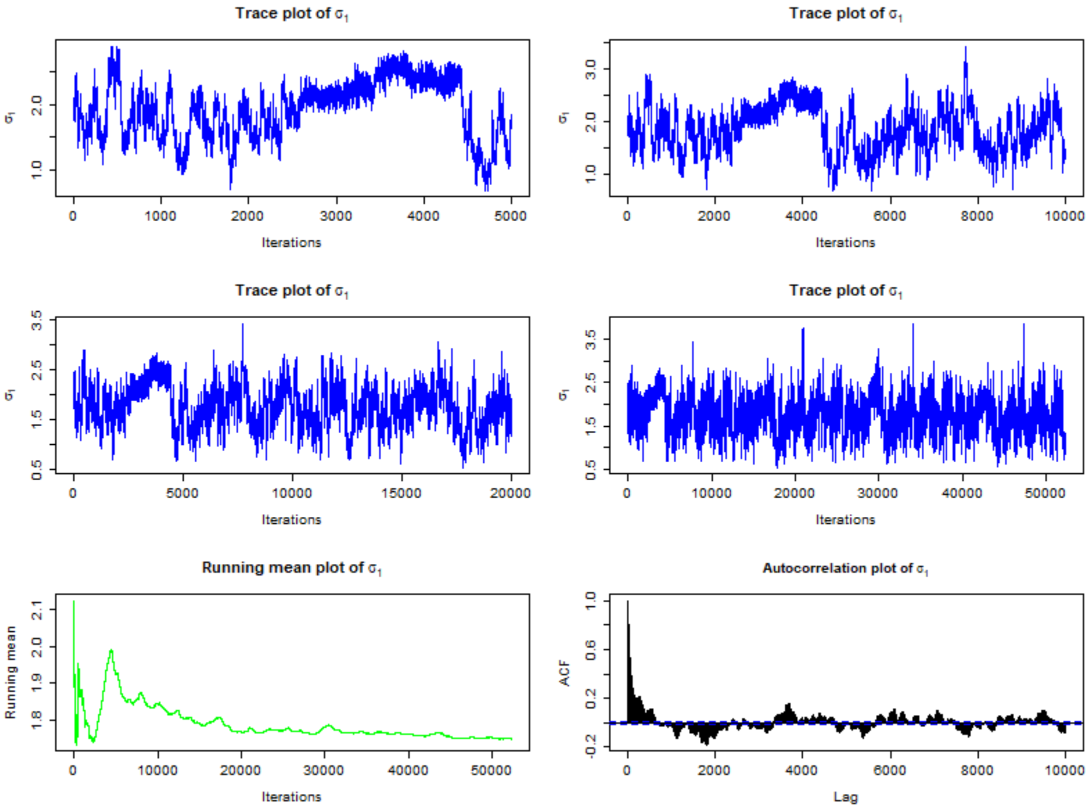

Figure 24: Plots for the parameter σ_1 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

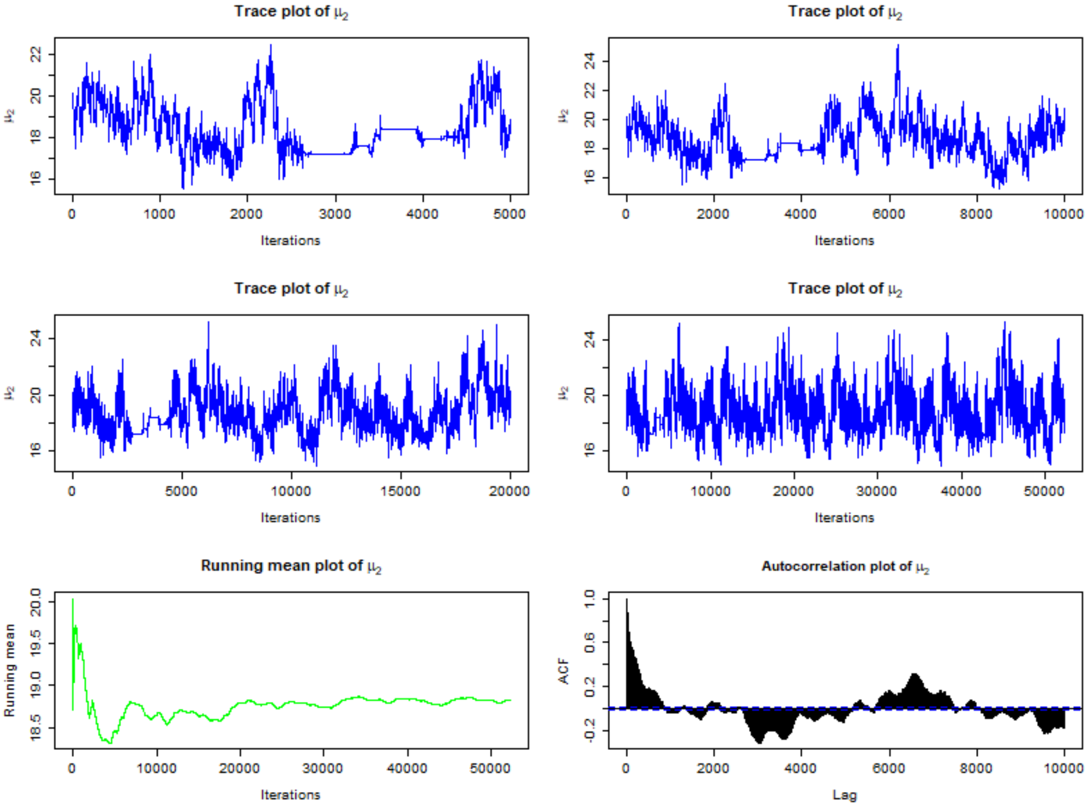

Figure 25: Plots for the parameter μ_2 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

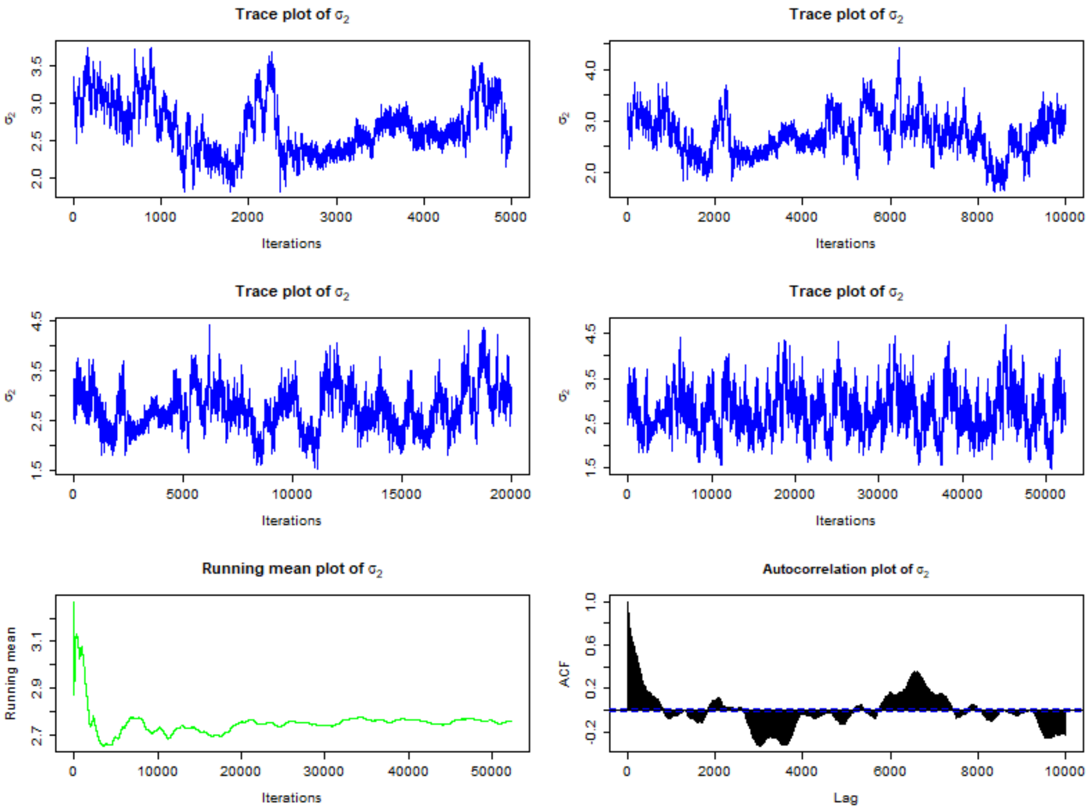

Figure 26: Plots for the parameter σ_2 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

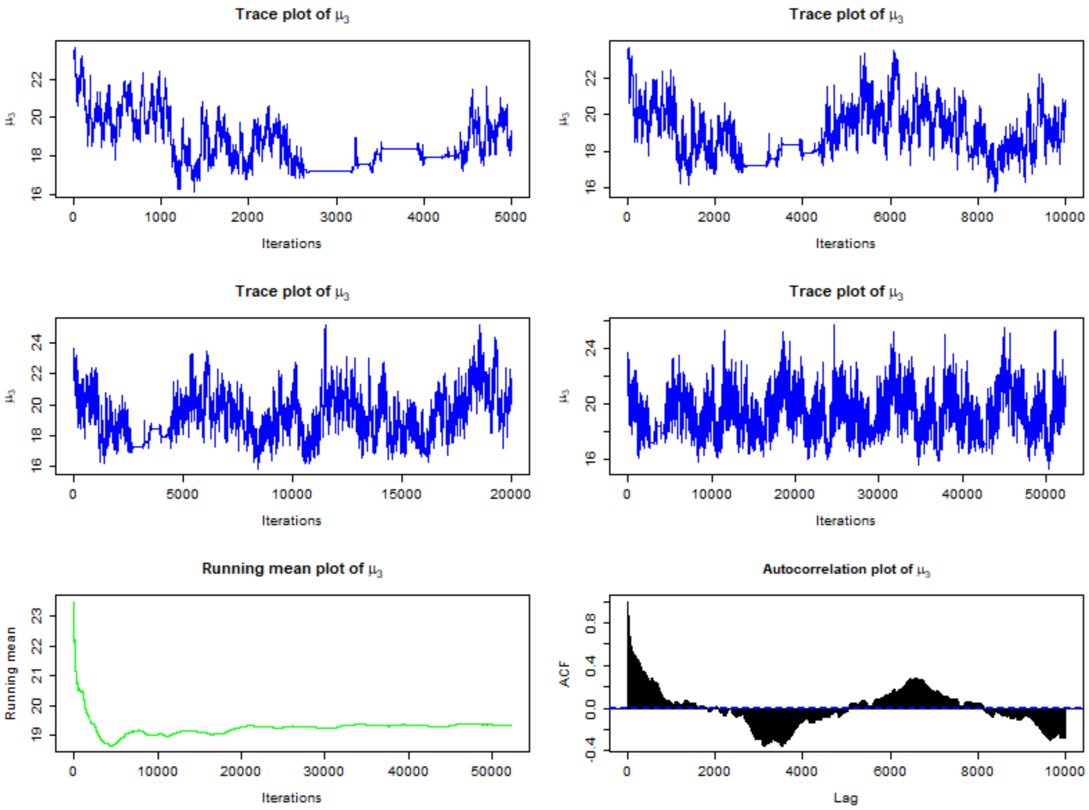

Figure 27: Plots for the parameter μ_3 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

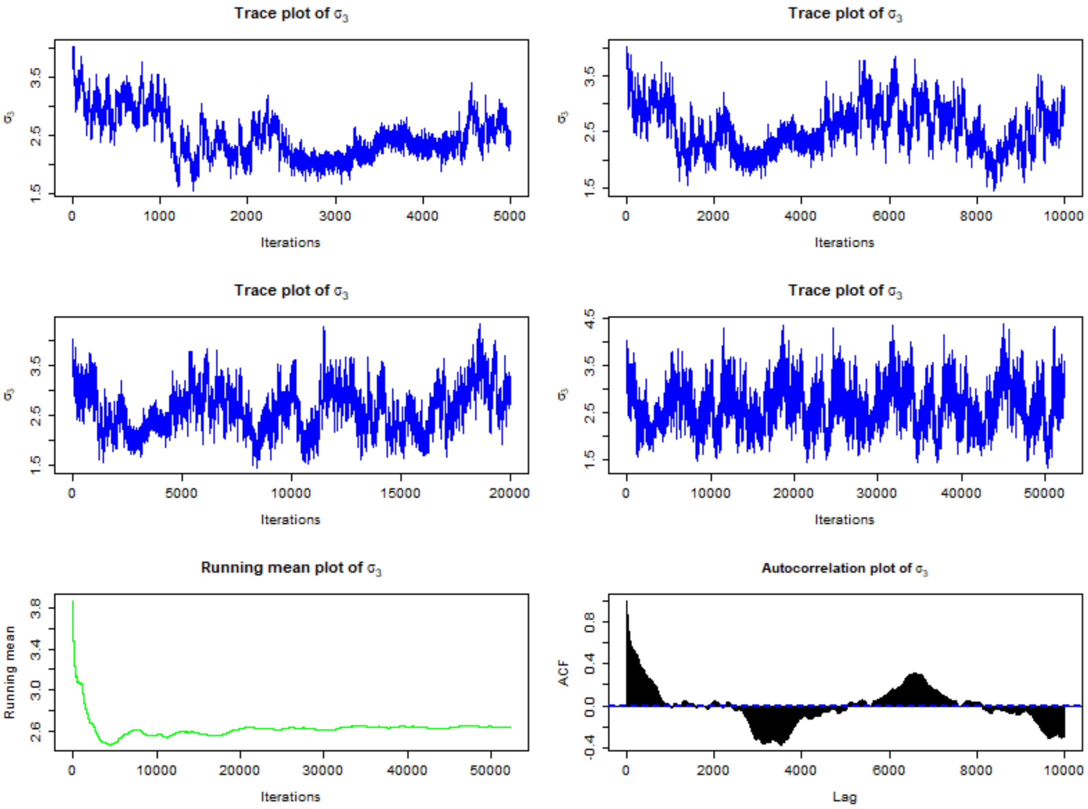

Figure 28: Plots for the parameter σ_3 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

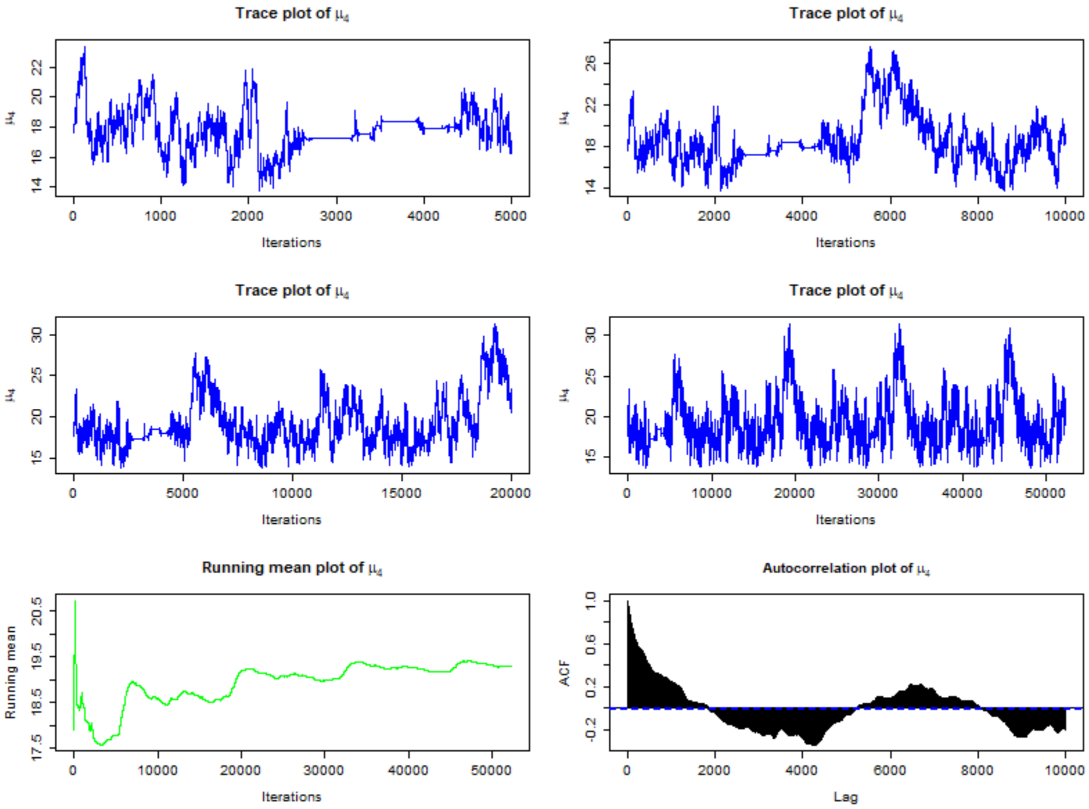

Figure 29: Plots for the parameter μ_4 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

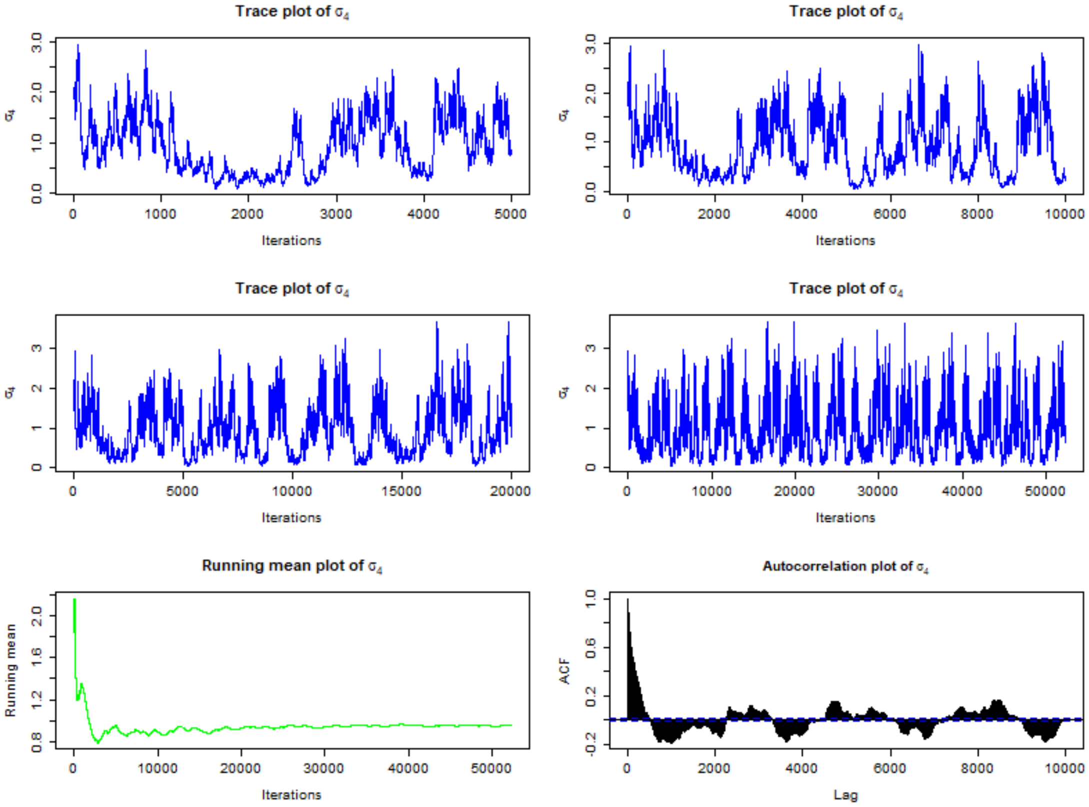

Figure 30: Plots for the parameter σ_4 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

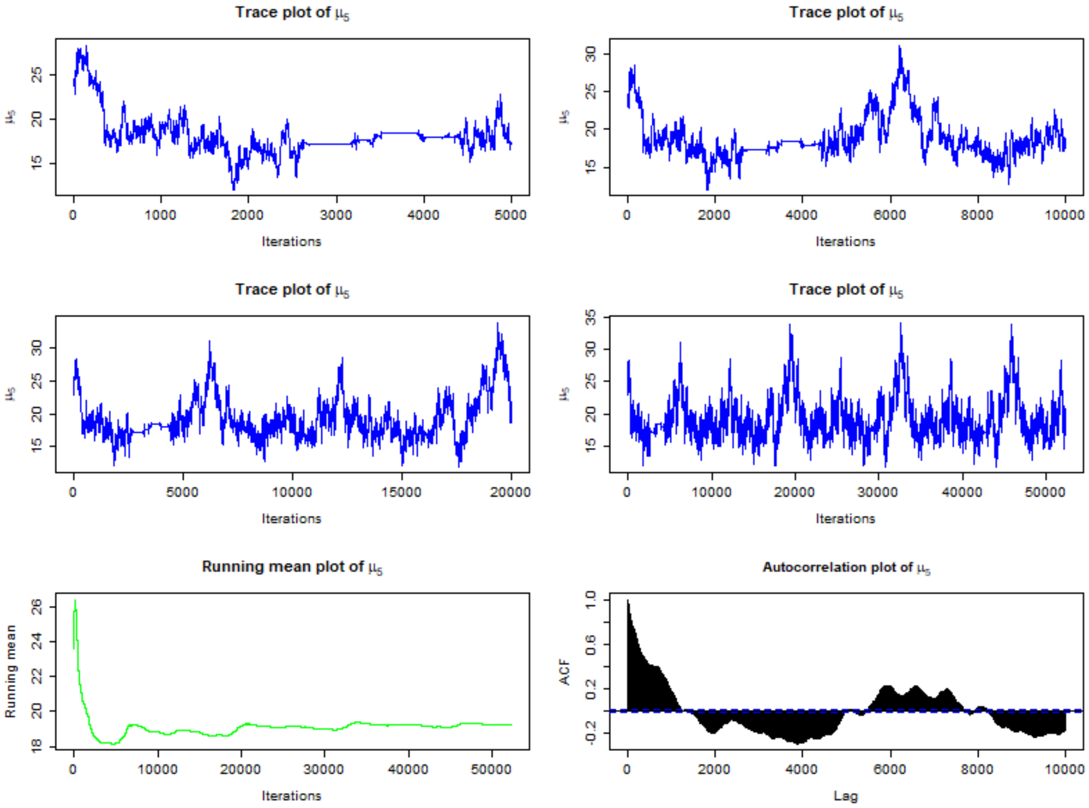

Figure 31: Plots for the parameter μ_5 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

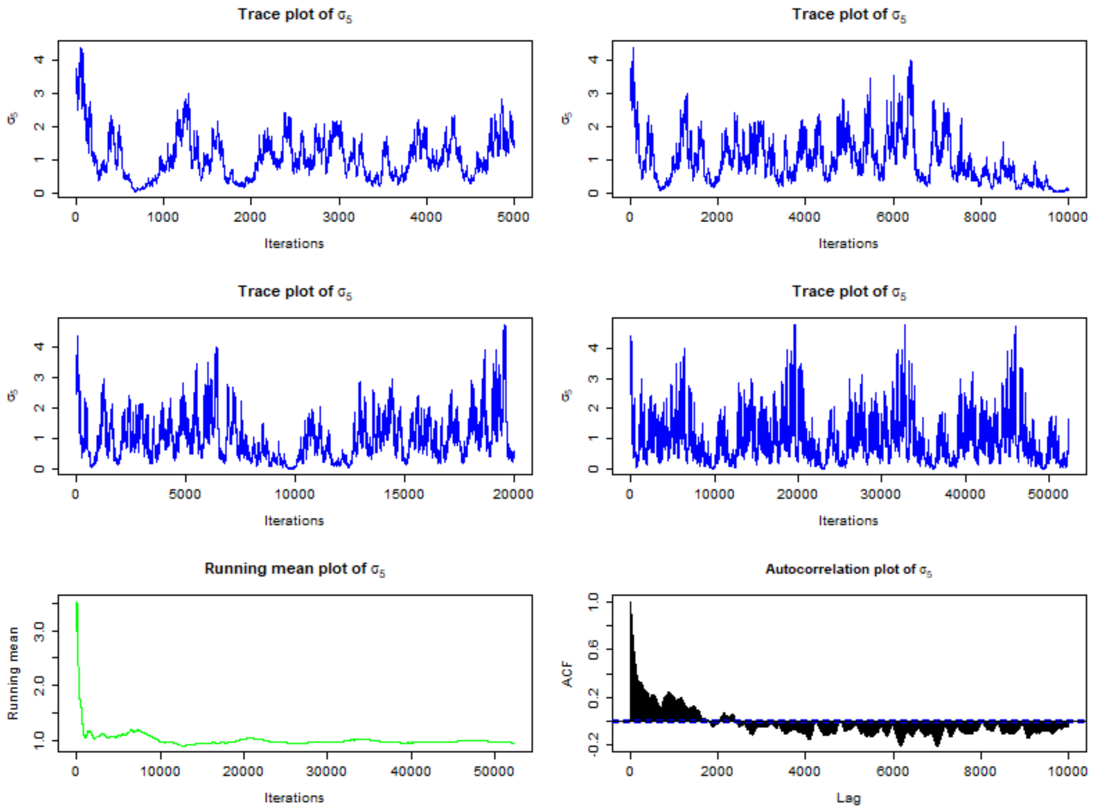

Figure 32: Plots for the parameter σ_5 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

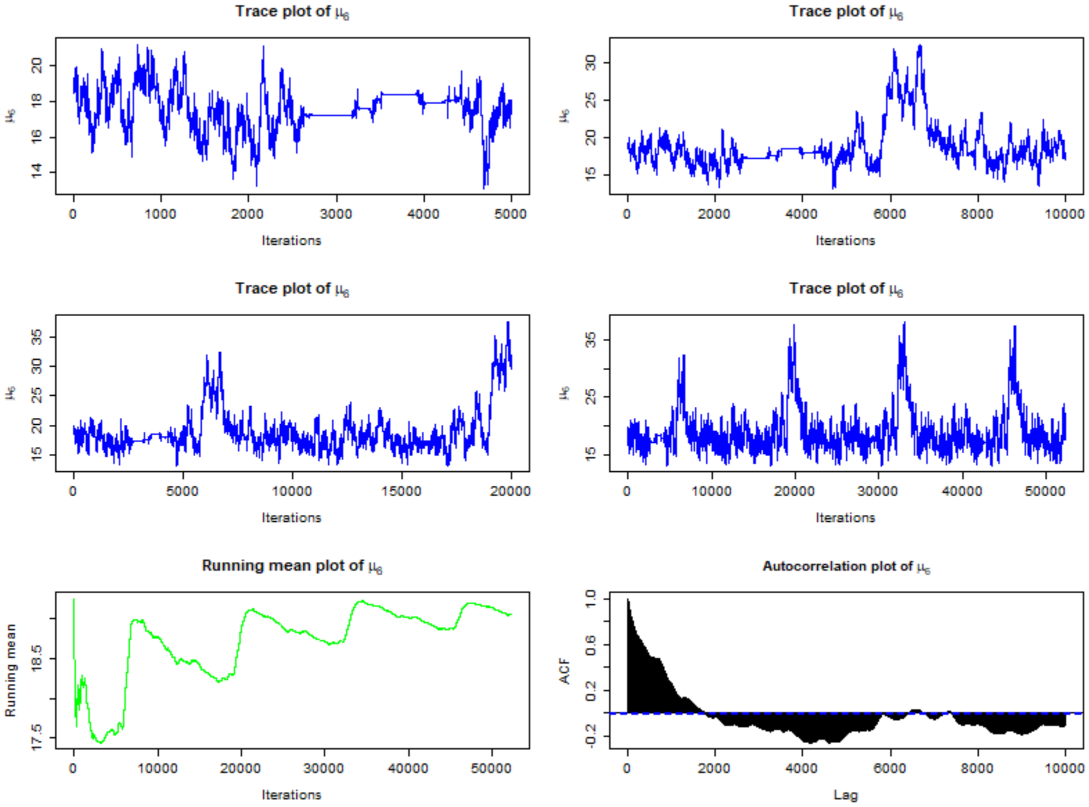

Figure 33: Plots for the parameter μ_6 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

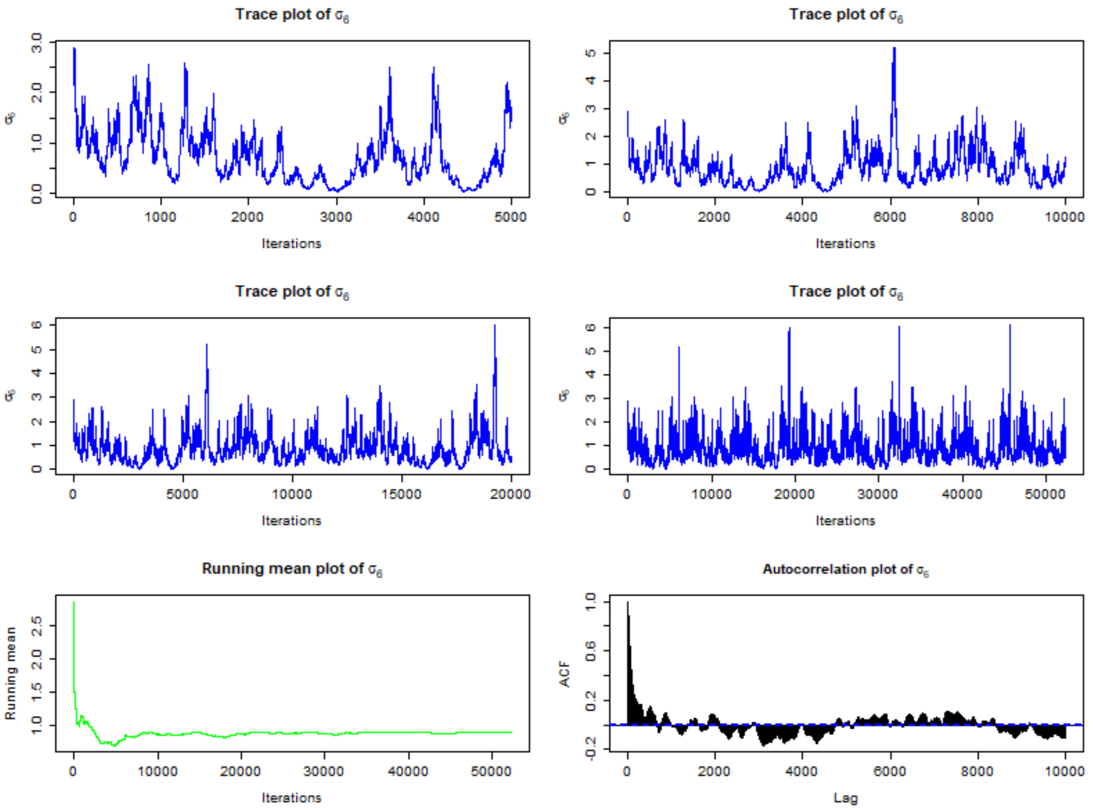

Figure 34: Plots for the parameter σ_6 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

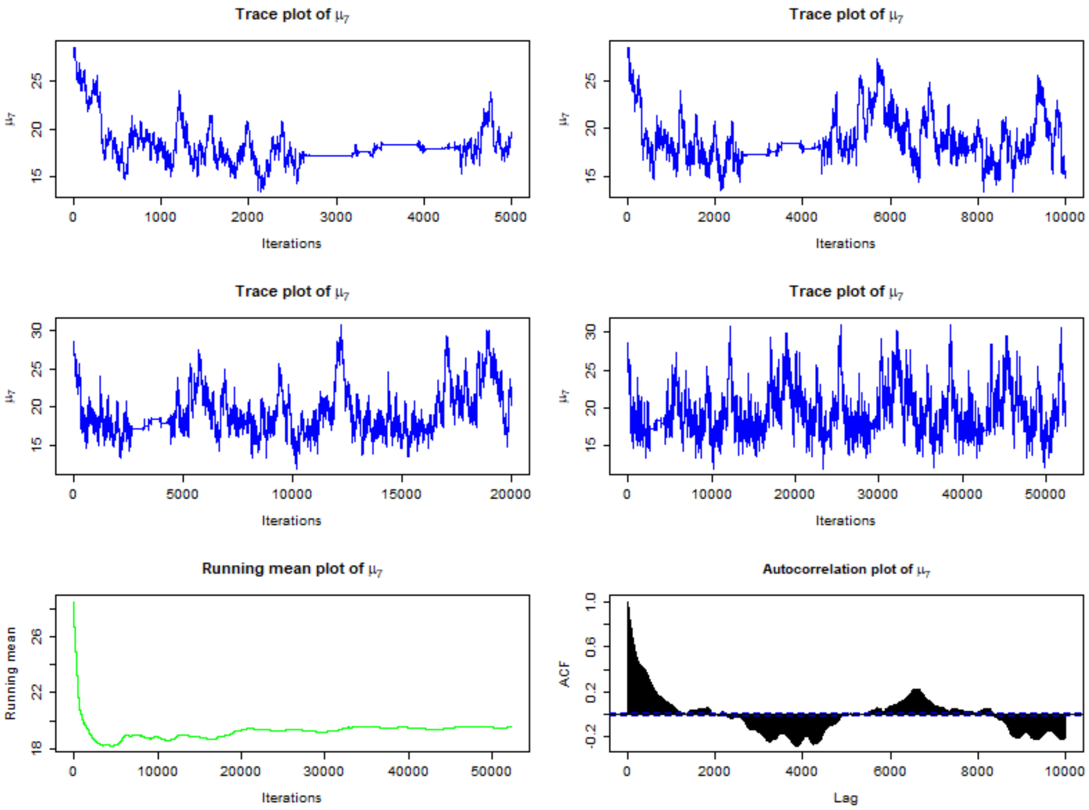

Figure 35: Plots for the parameter μ_7 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

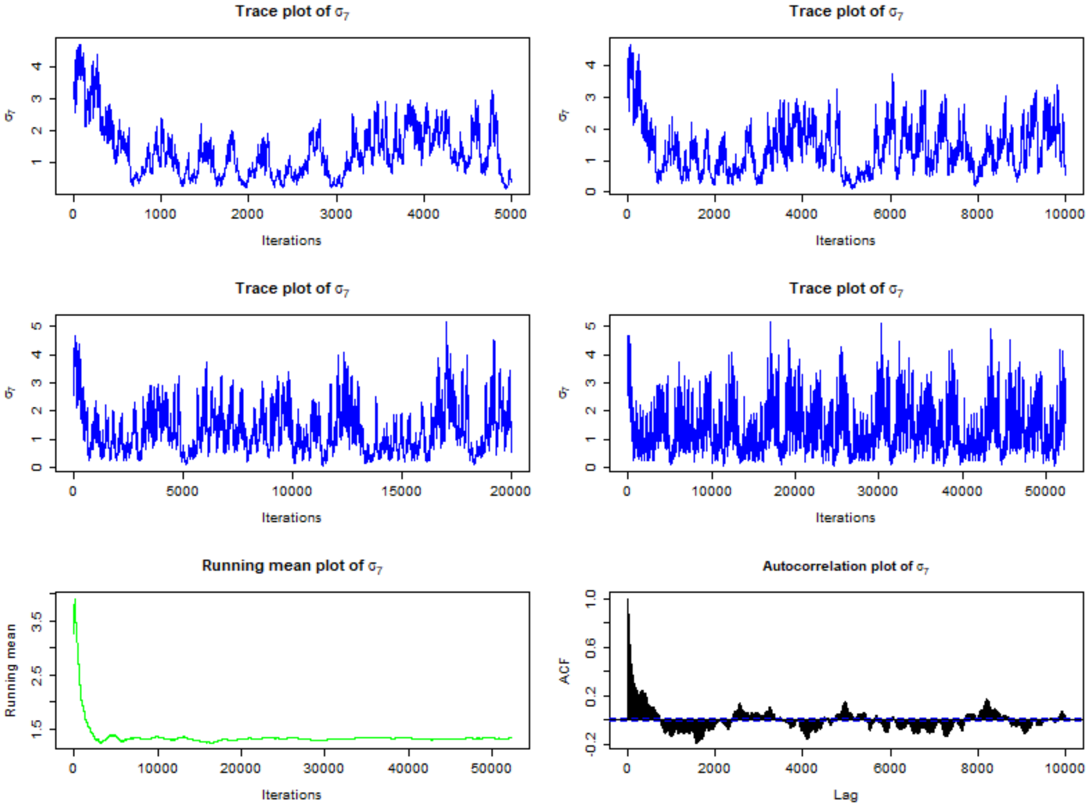

Figure 36: Plots for the parameter σ_7 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

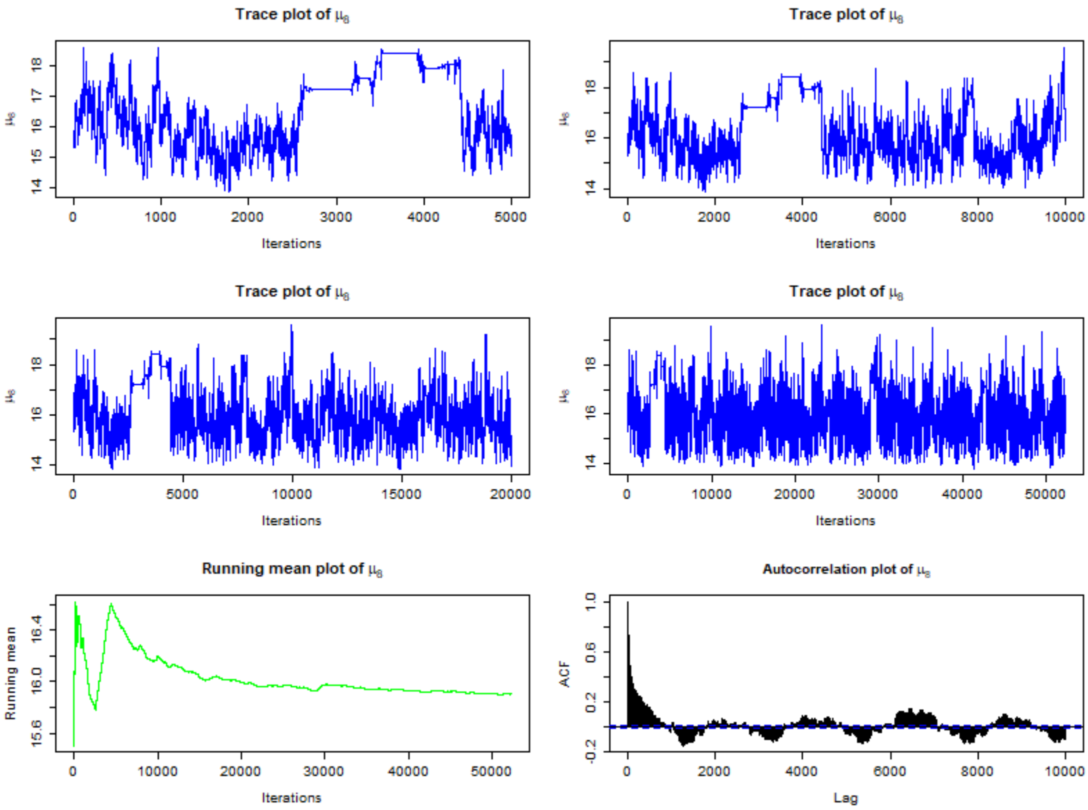

Figure 37: Plots for the parameter μ_8 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

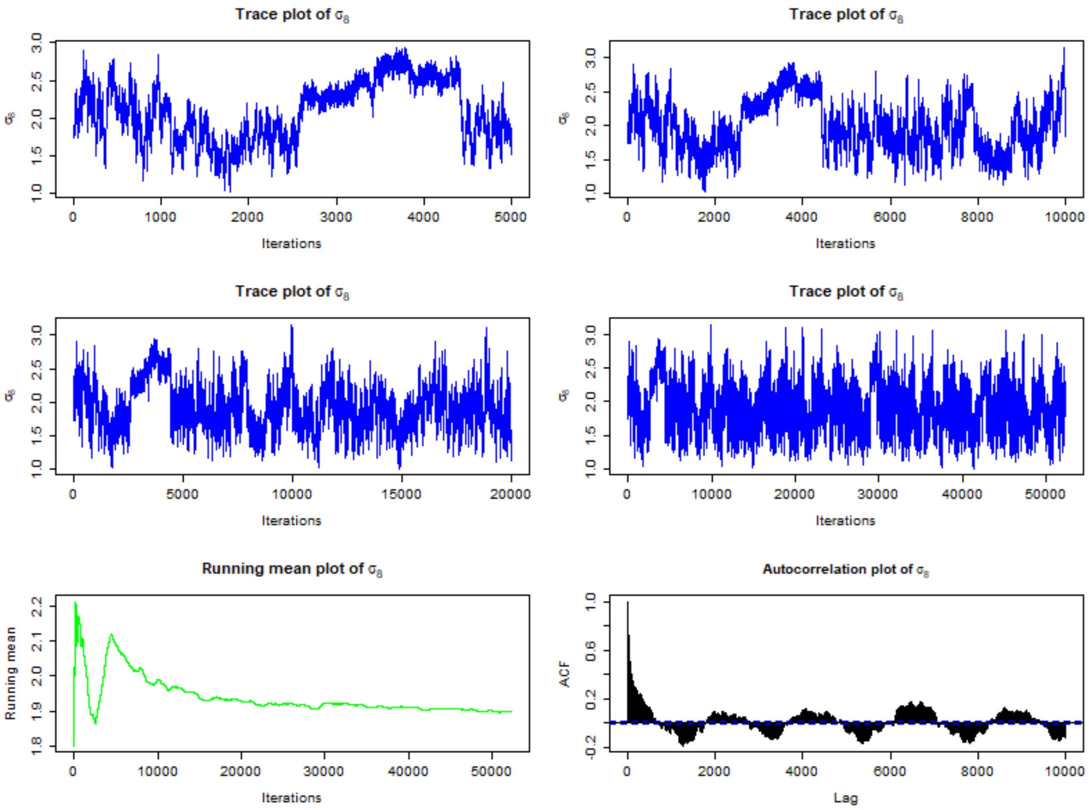

Figure 38: Plots for the parameter σ_8 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

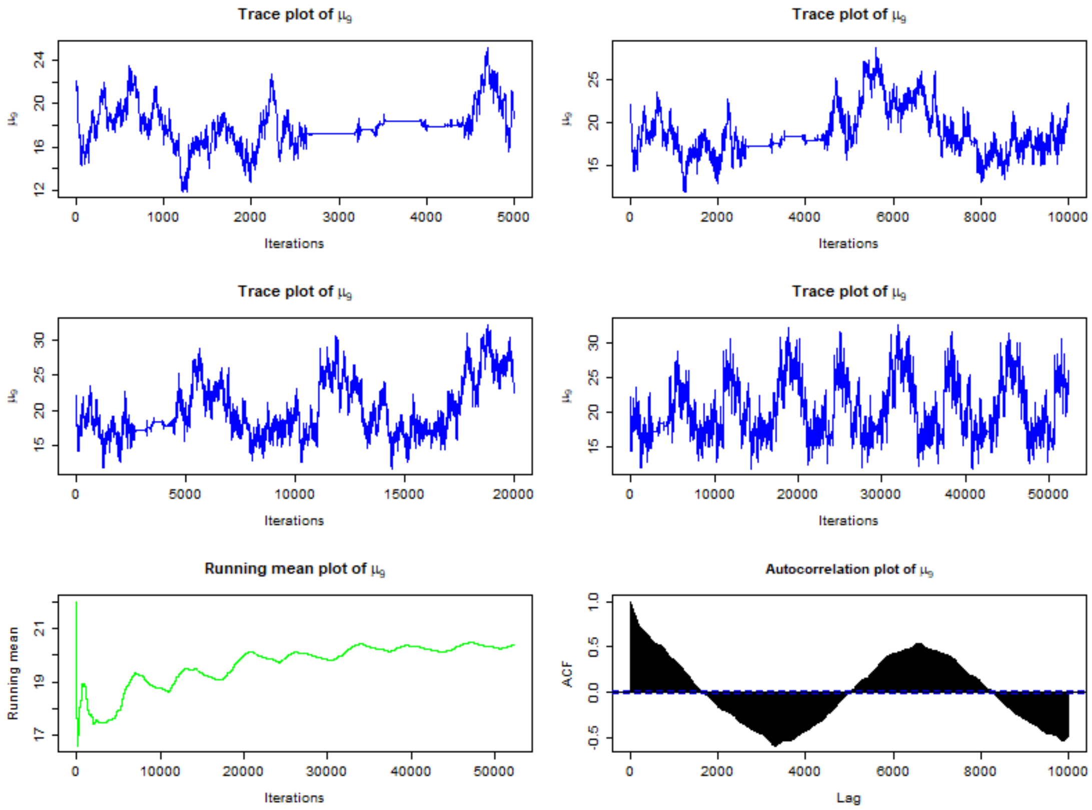

Figure 39: Plots for the parameter μ_9 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

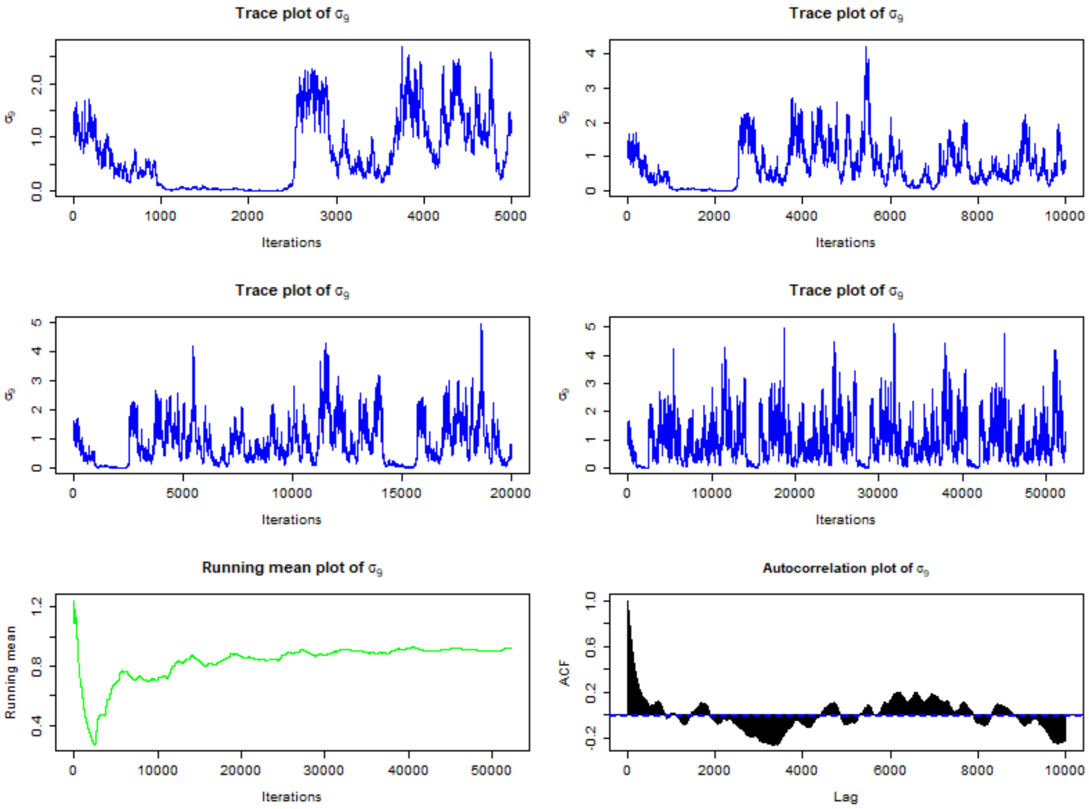

Figure 40: Plots for the parameter σ_9 based on the MCMC run for the segment-wise model. The two upper rows show trace plots for the first 5000, 10000 and 20000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

Line-wise model

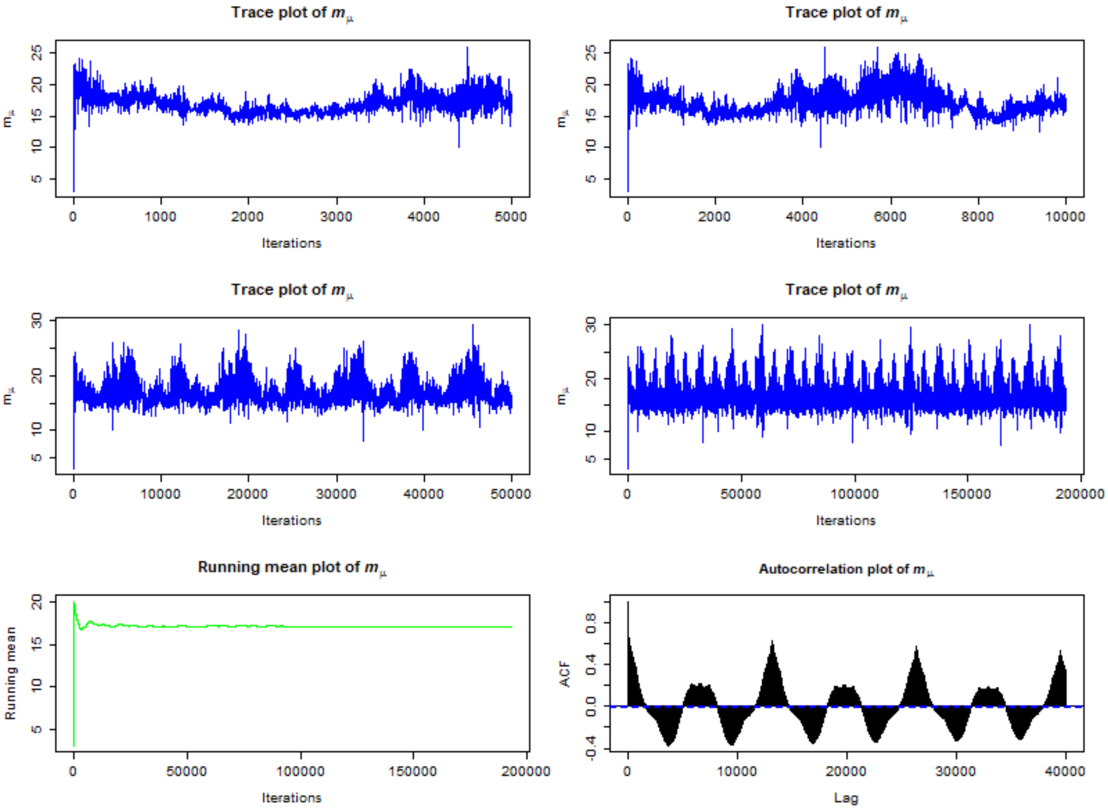

Figure 41: Plots for the parameter m_μ based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

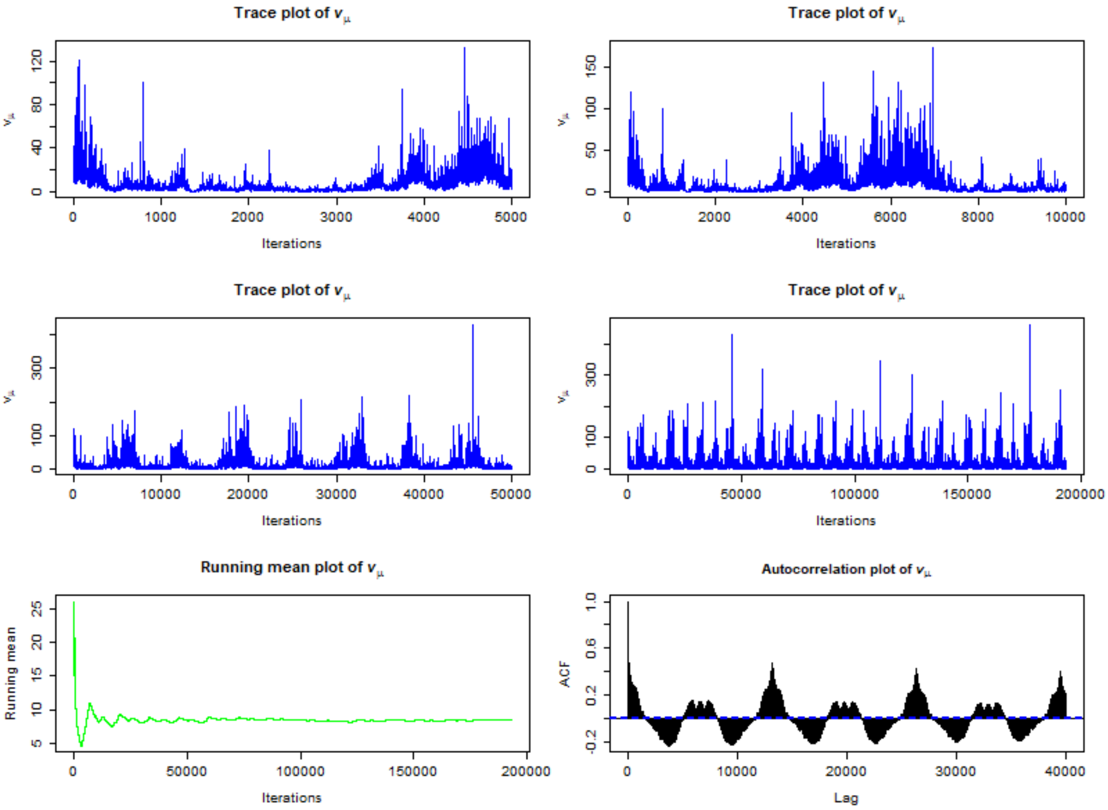

Figure 42: Plots for the parameter v_μ based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

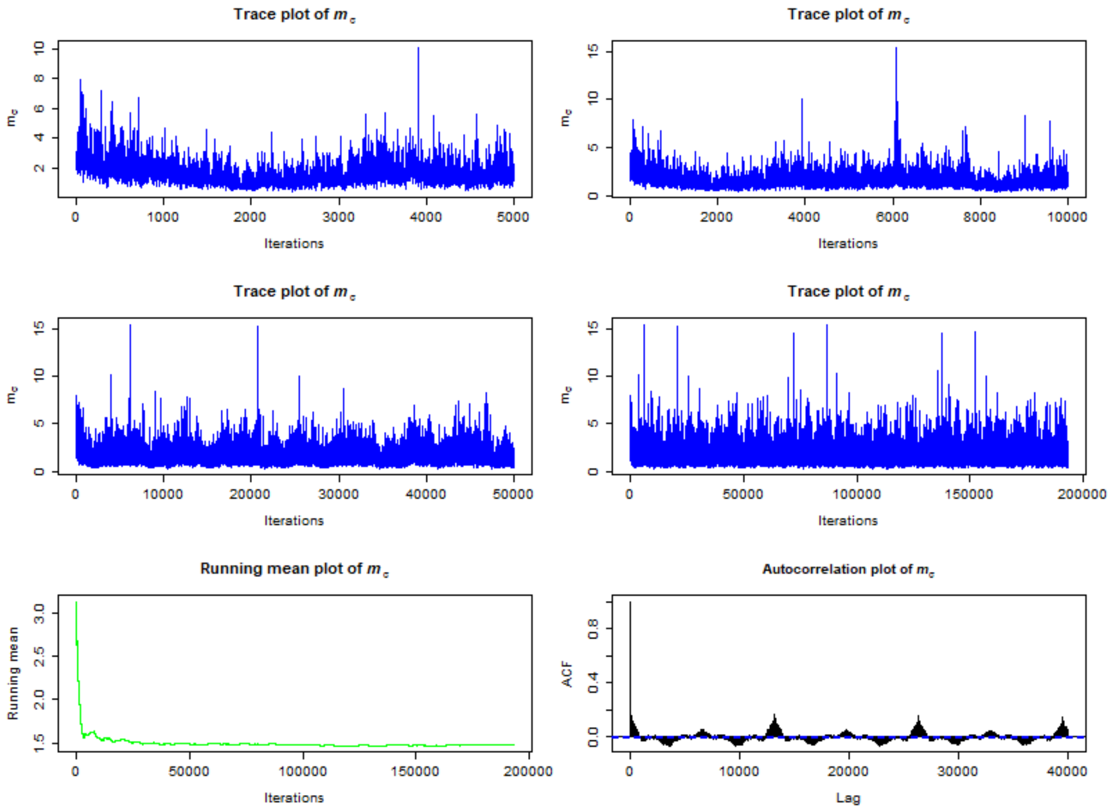

Figure 43: Plots for the parameter m_σ based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

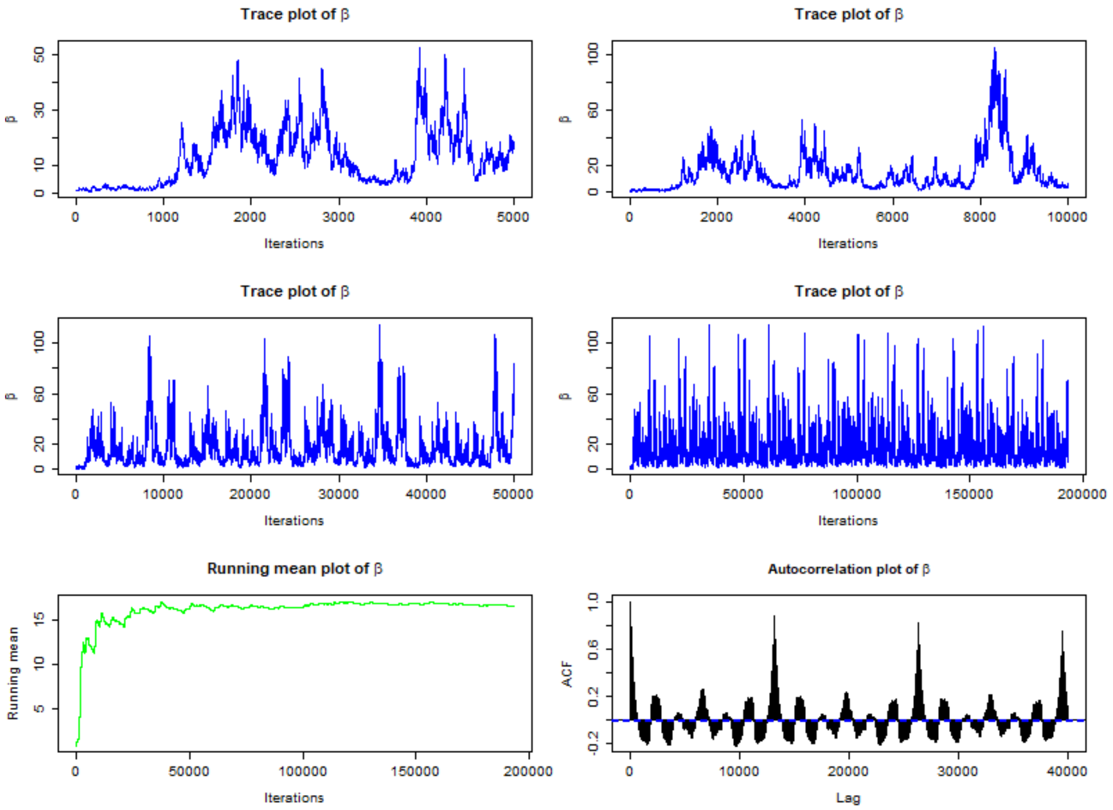

Figure 44: Plots for the parameter β based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

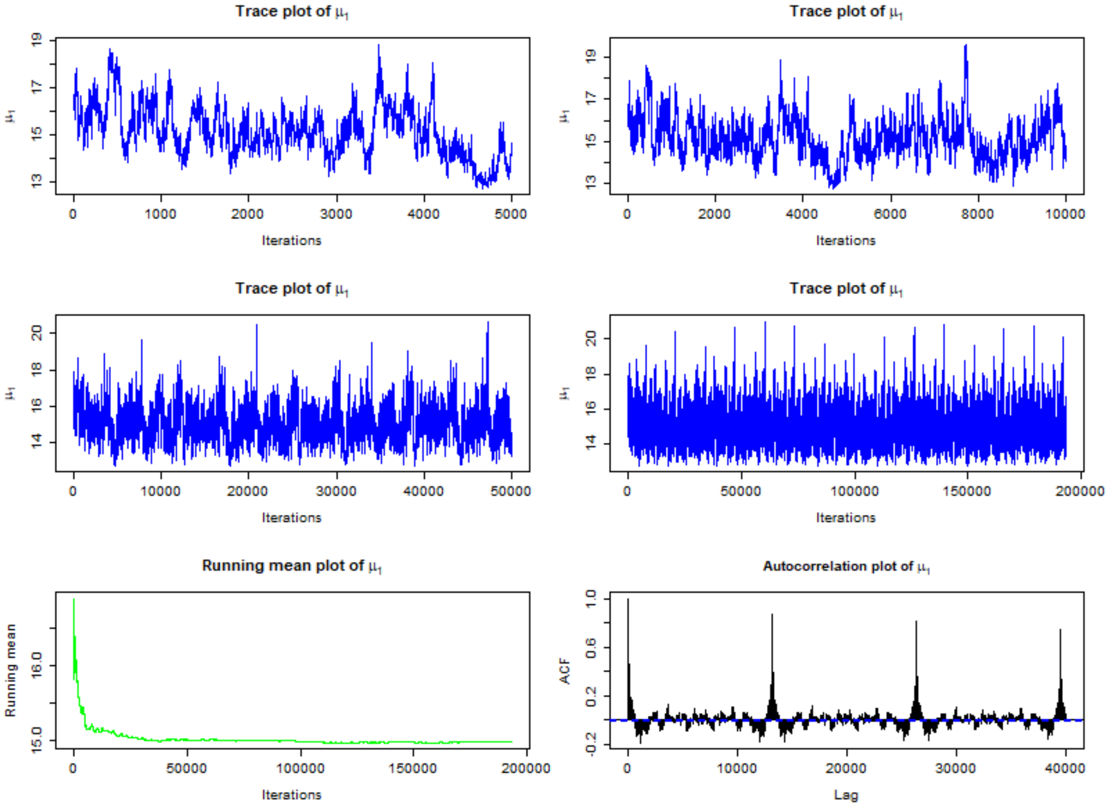

Figure 45: Plots for the parameter μ_1 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

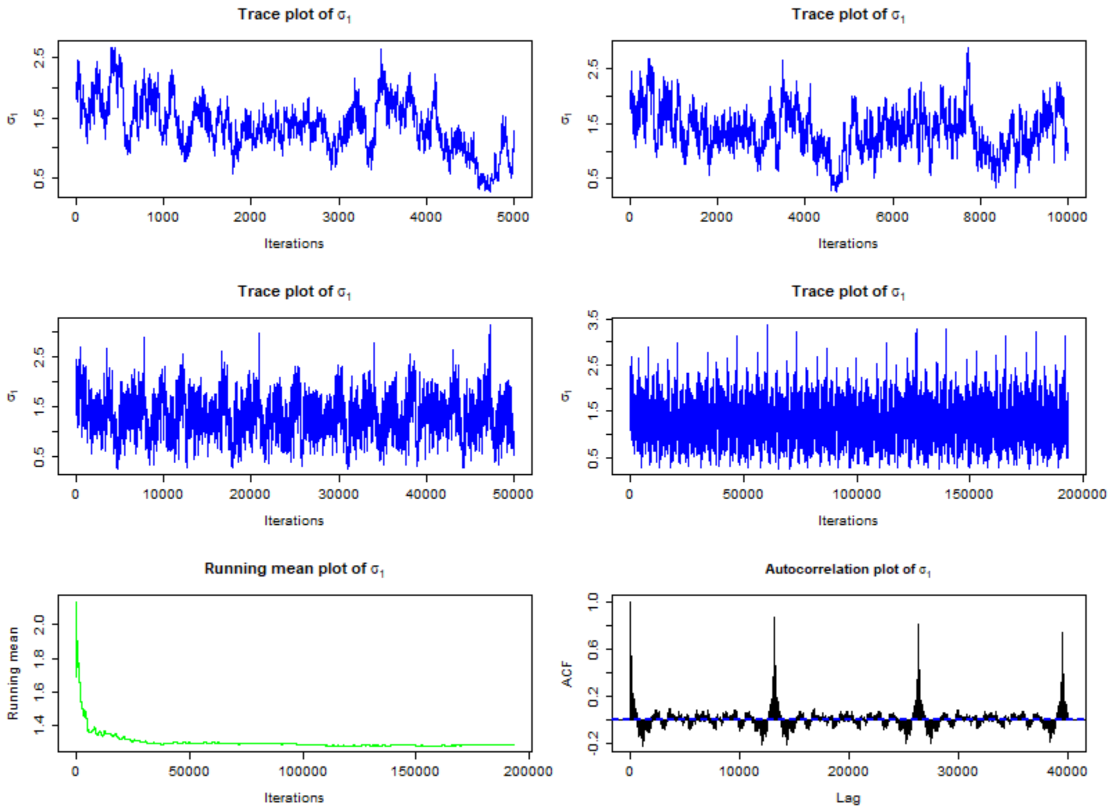

Figure 46: Plots for the parameter σ_1 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

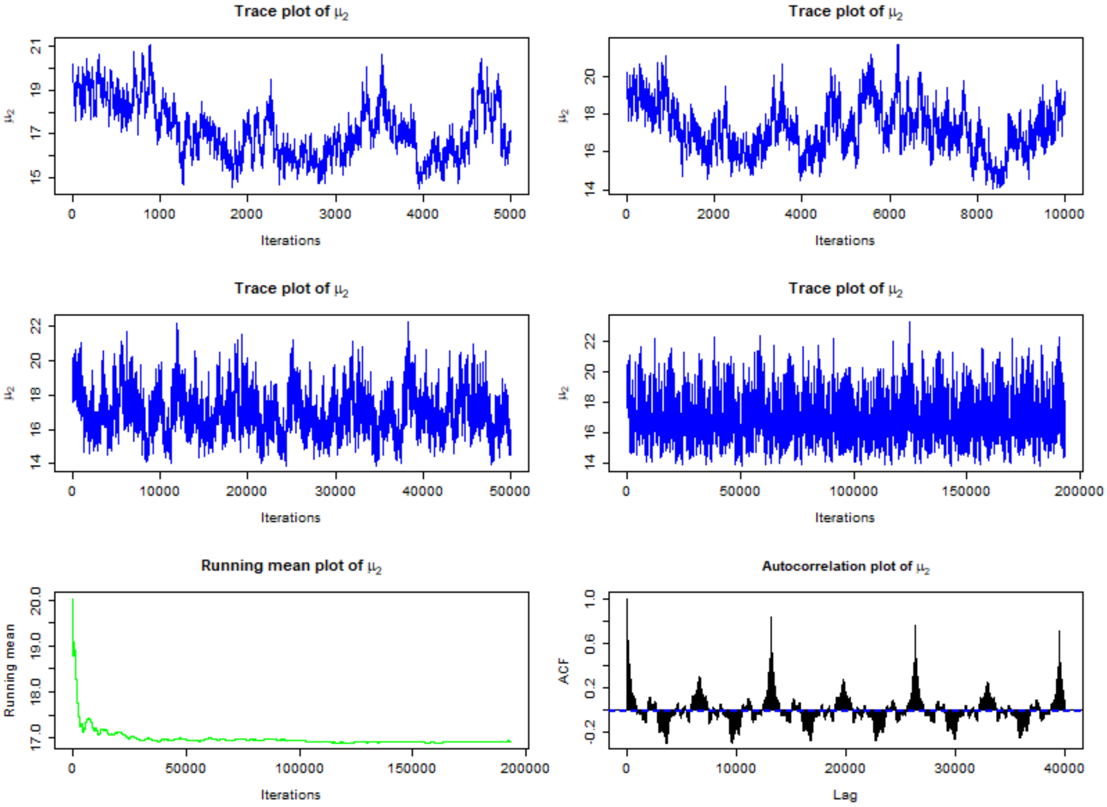

Figure 47: Plots for the parameter μ_2 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

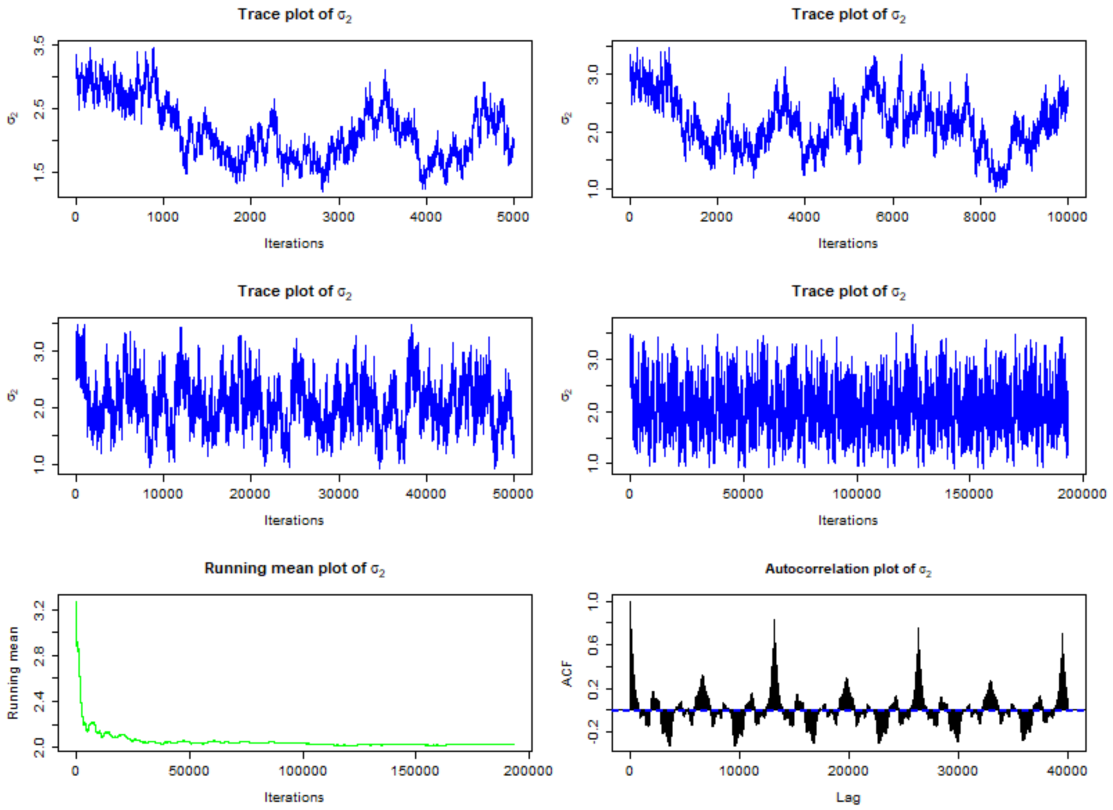

Figure 48: Plots for the parameter σ_2 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

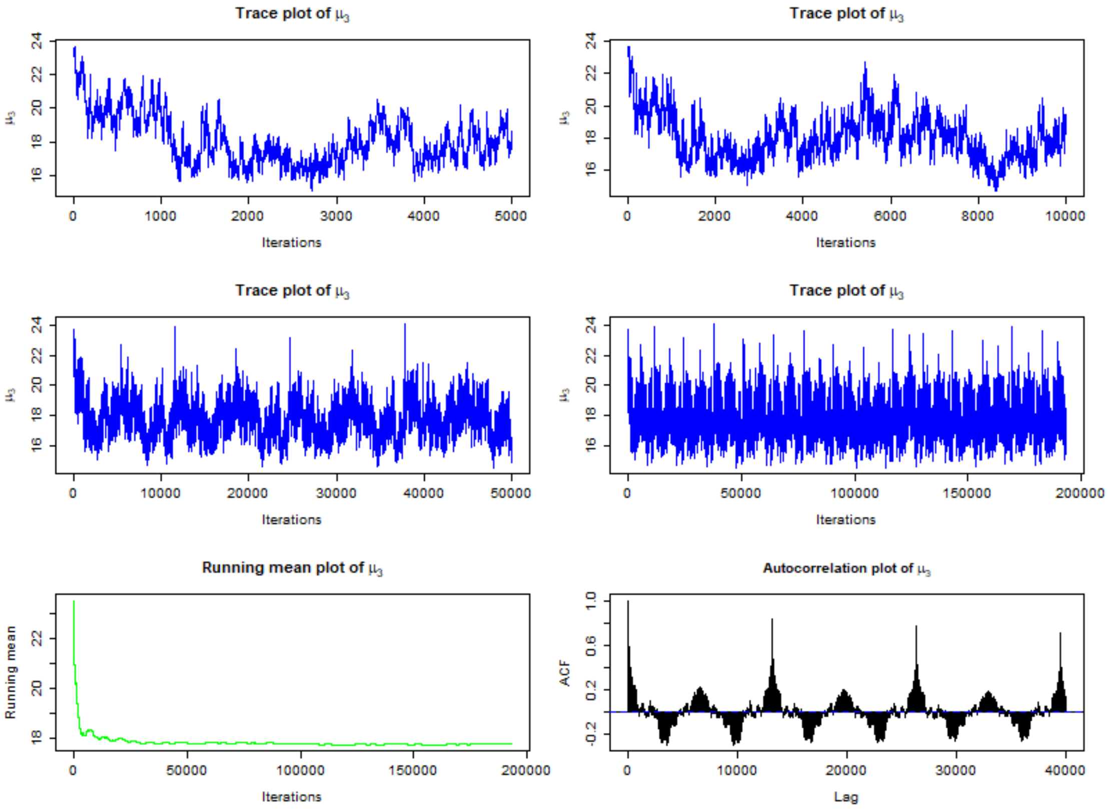

Figure 49: Plots for the parameter μ_3 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

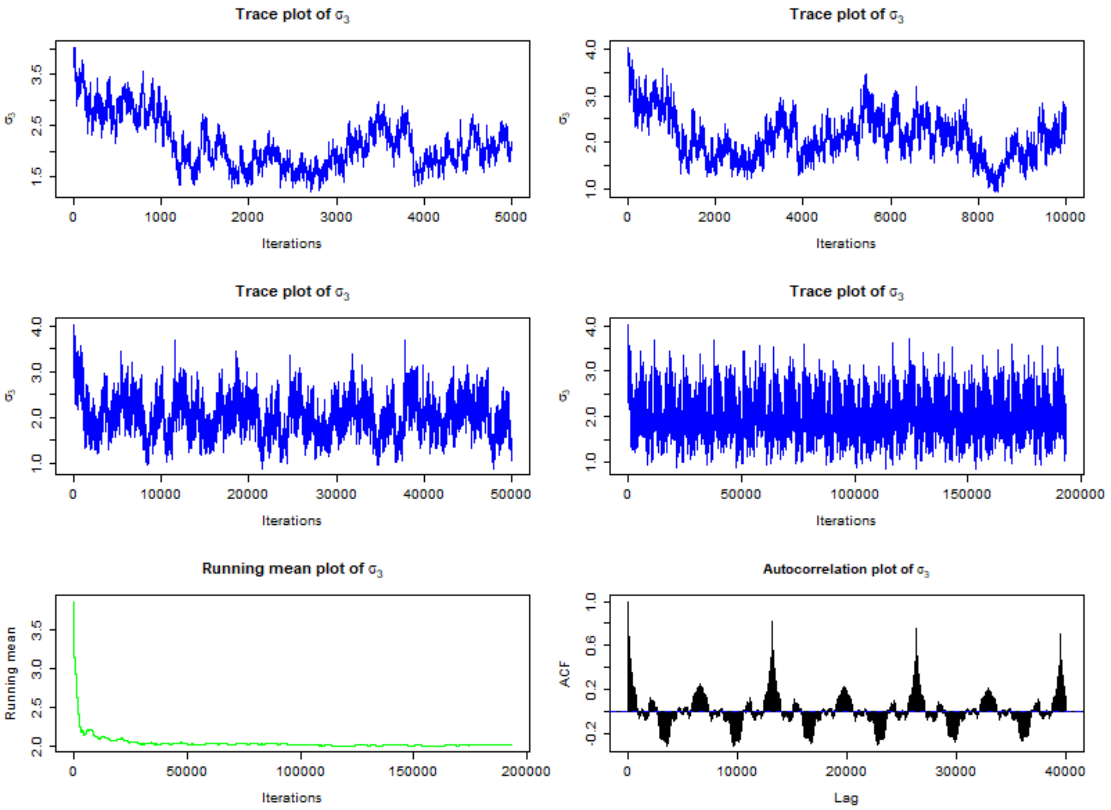

Figure 50: Plots for the parameter σ_3 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

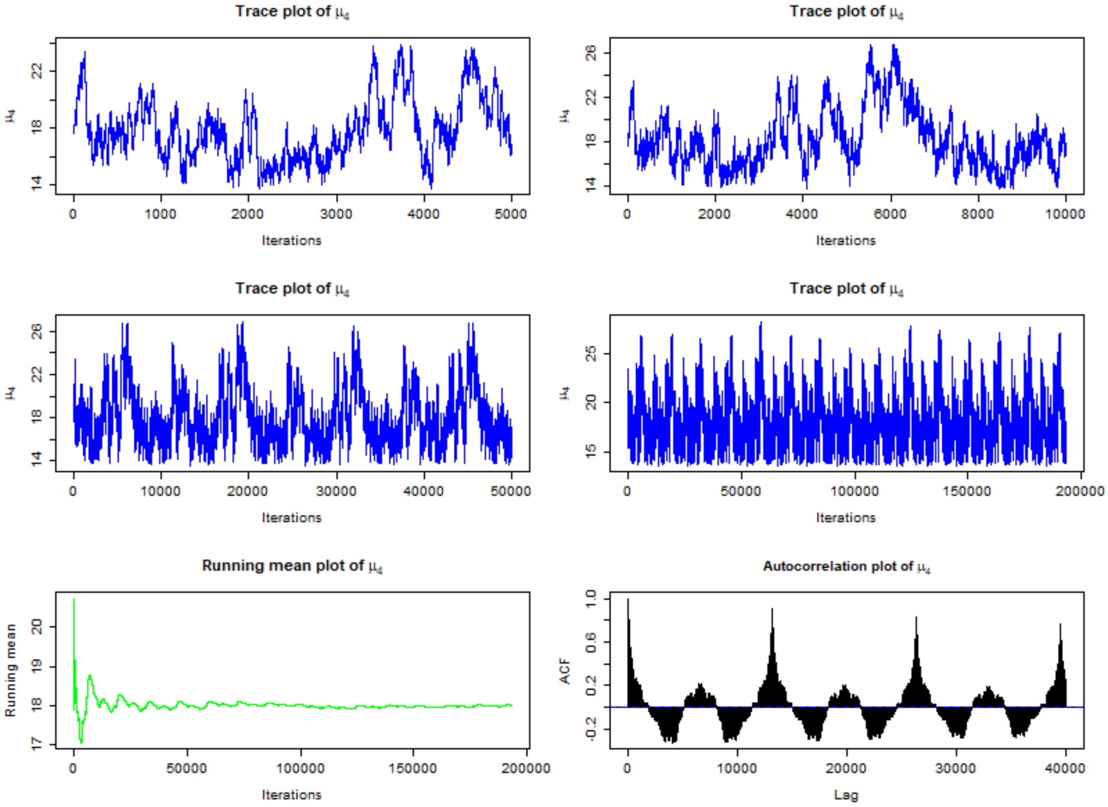

Figure 51: Plots for the parameter μ_4 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

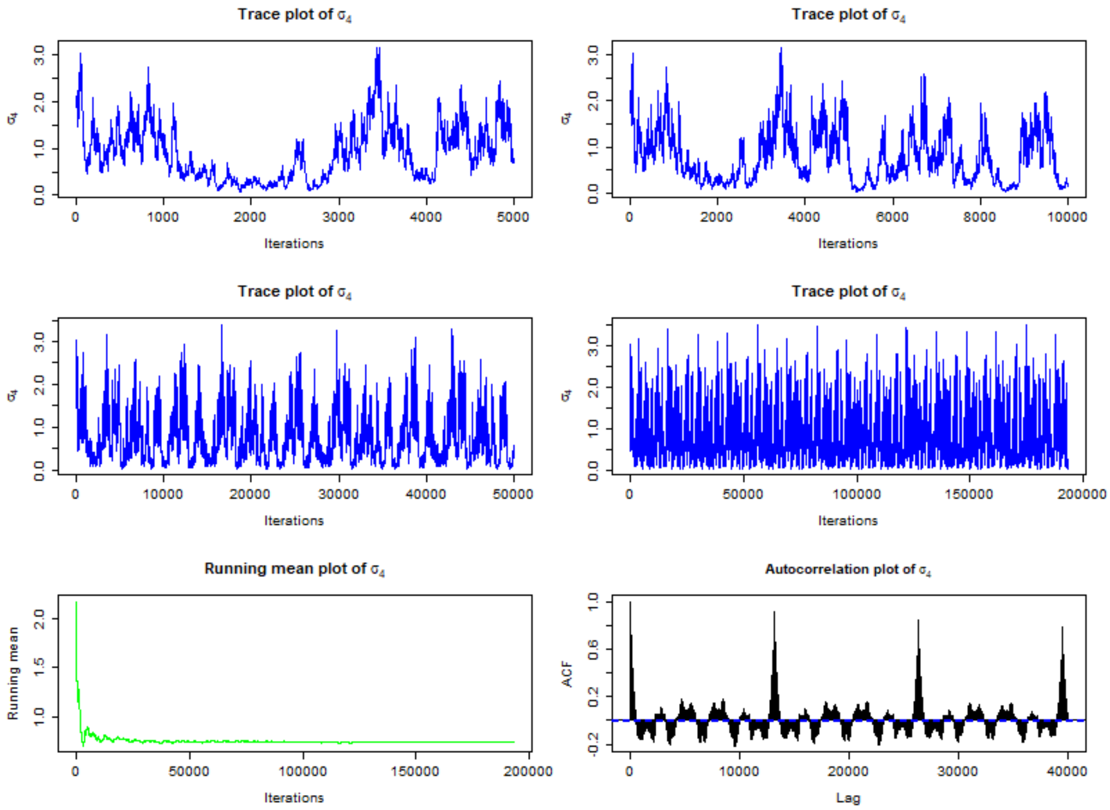

Figure 52: Plots for the parameter σ_4 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

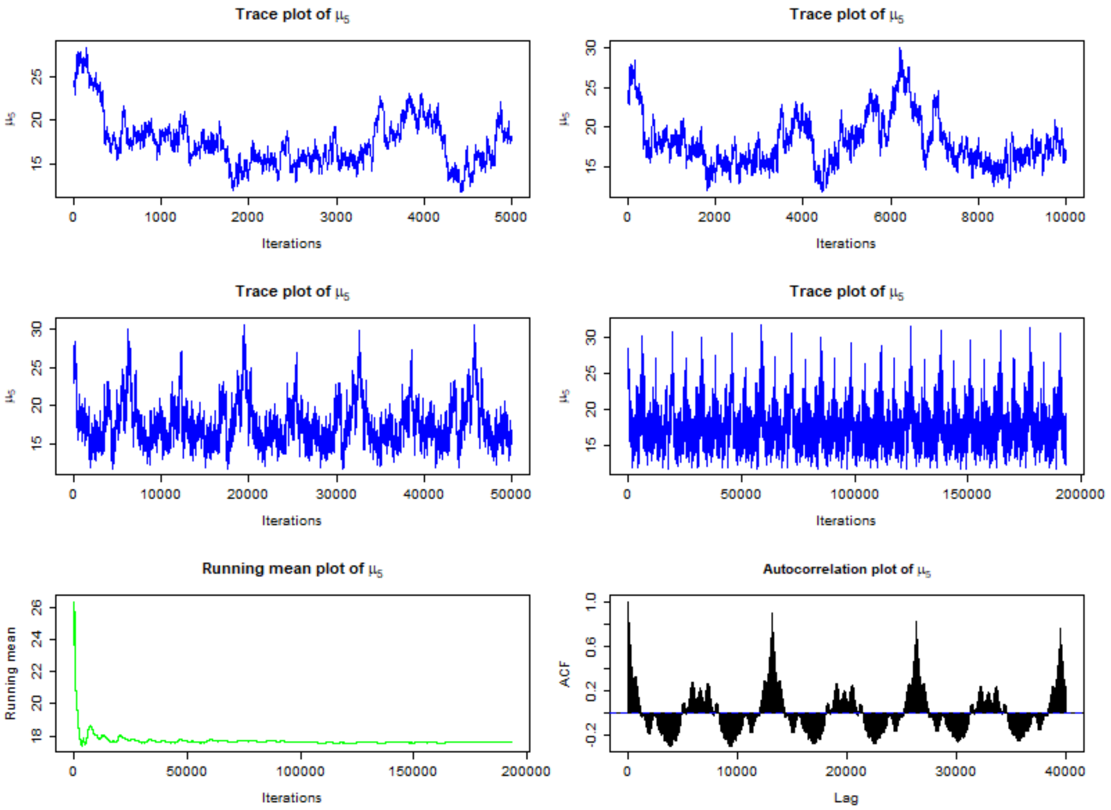

Figure 53: Plots for the parameter μ_5 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

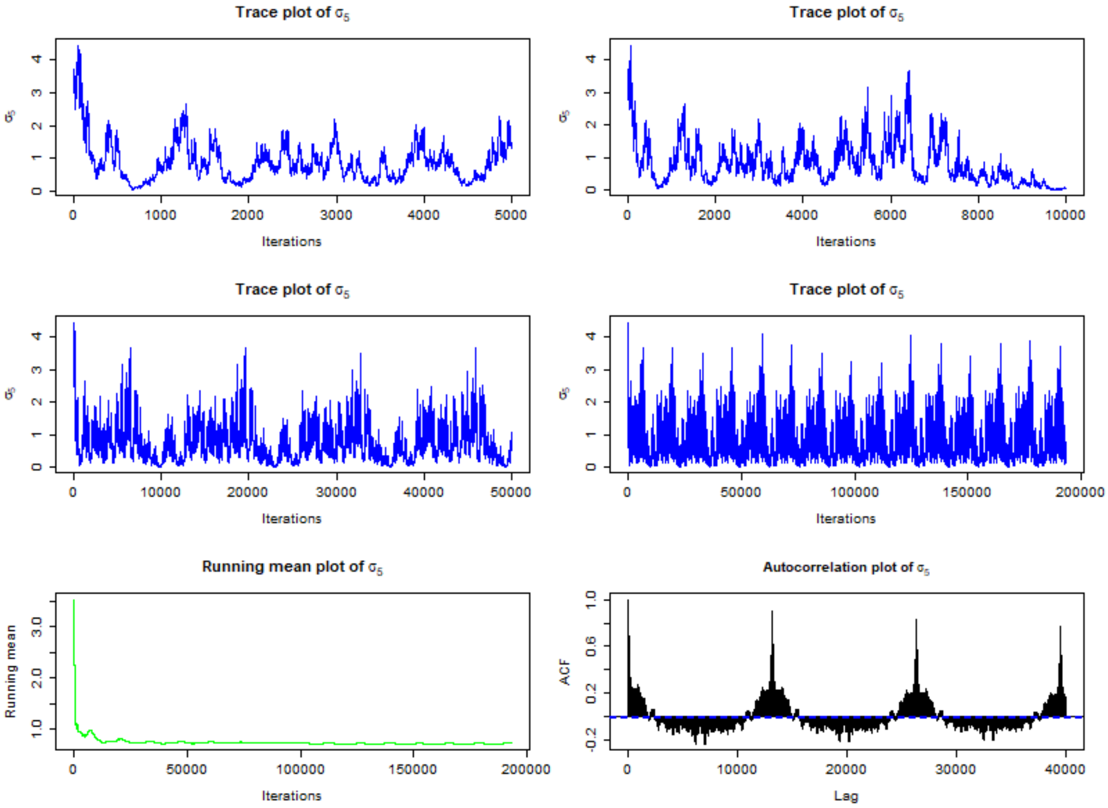

Figure 54: Plots for the parameter σ_5 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

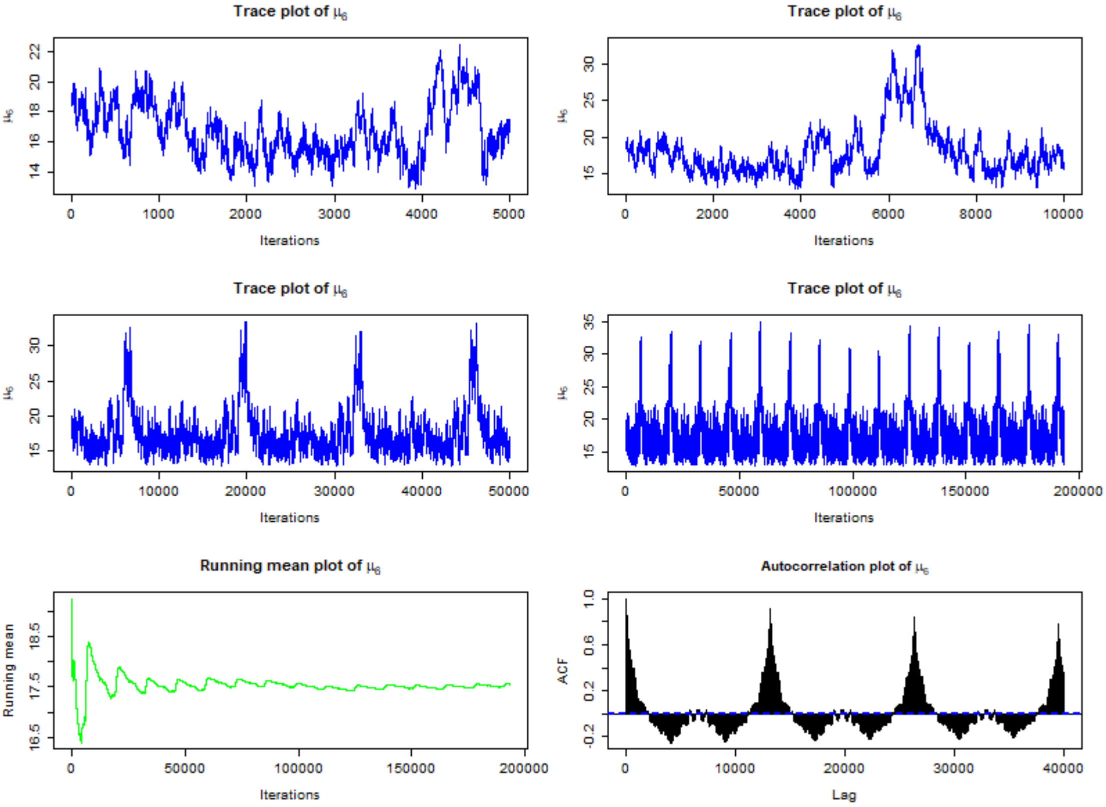

Figure 55: Plots for the parameter μ_6 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

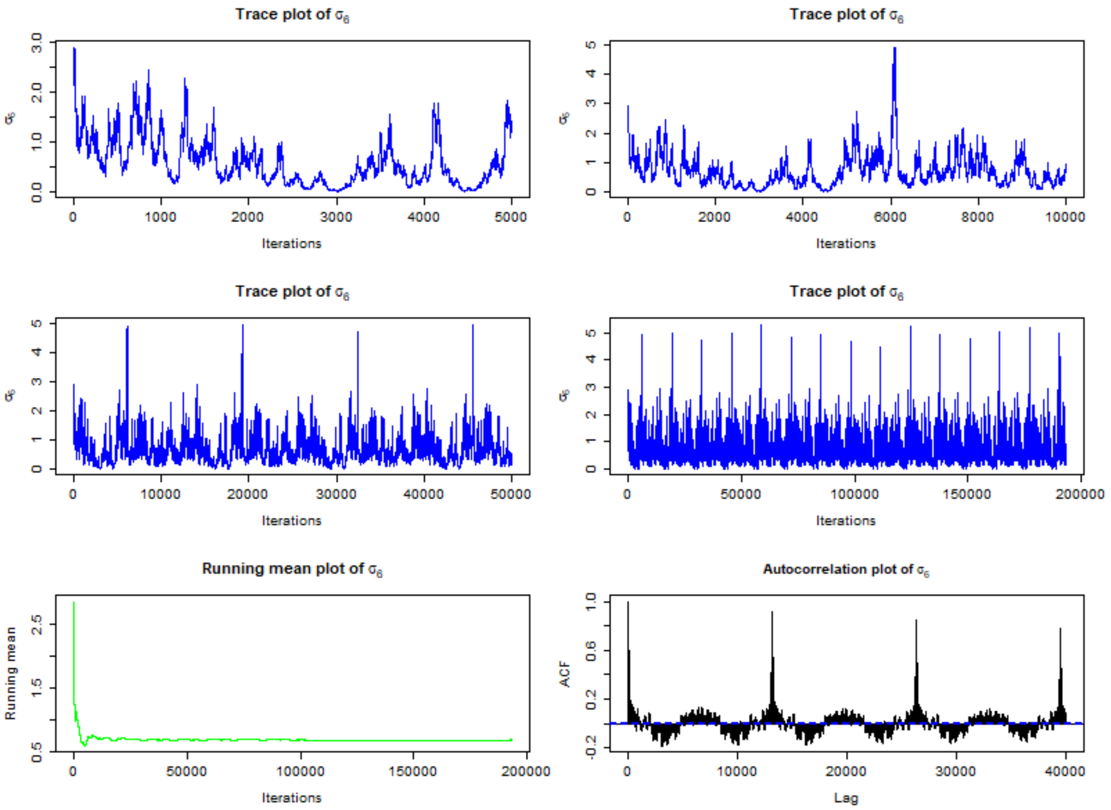

Figure 56: Plots for the parameter σ_6 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

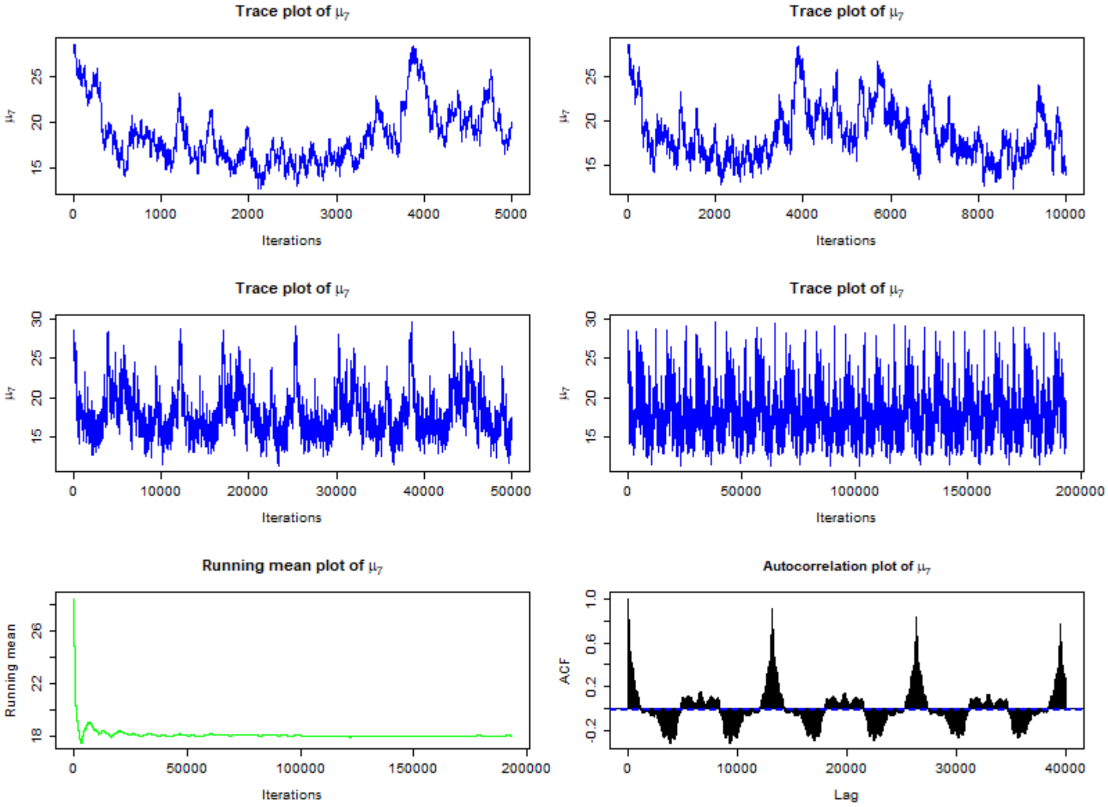

Figure 57: Plots for the parameter μ_7 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

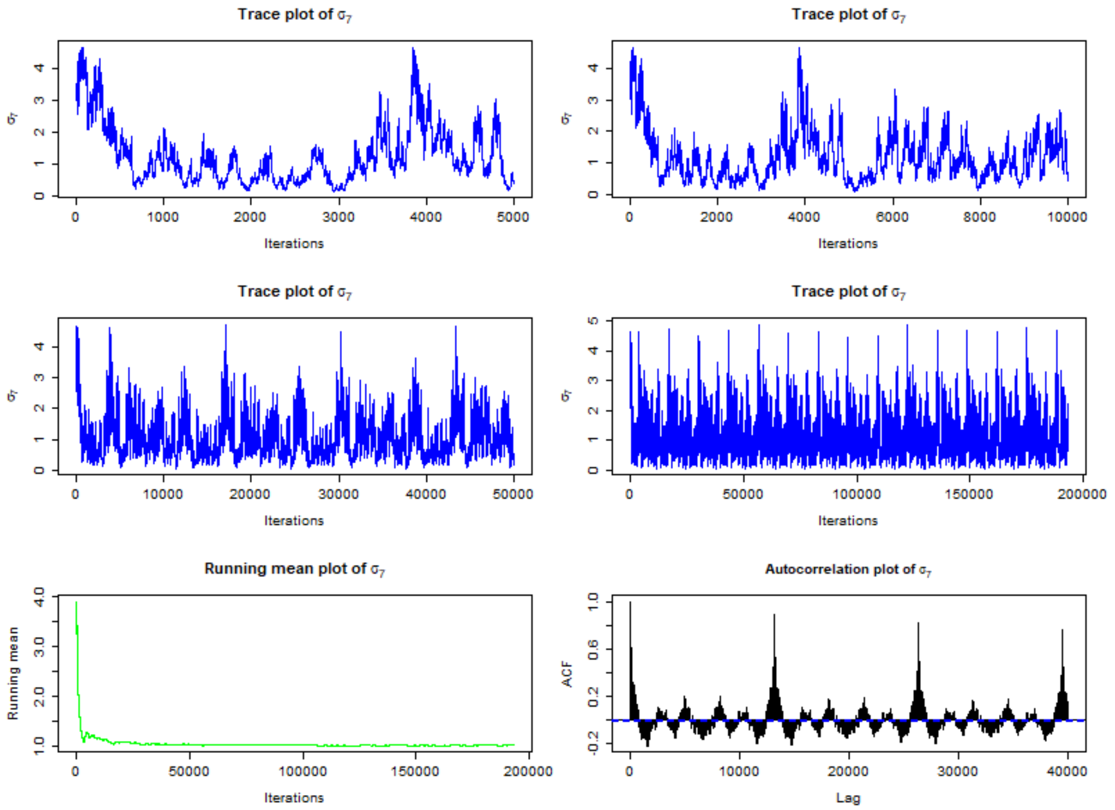

Figure 58: Plots for the parameter σ_7 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

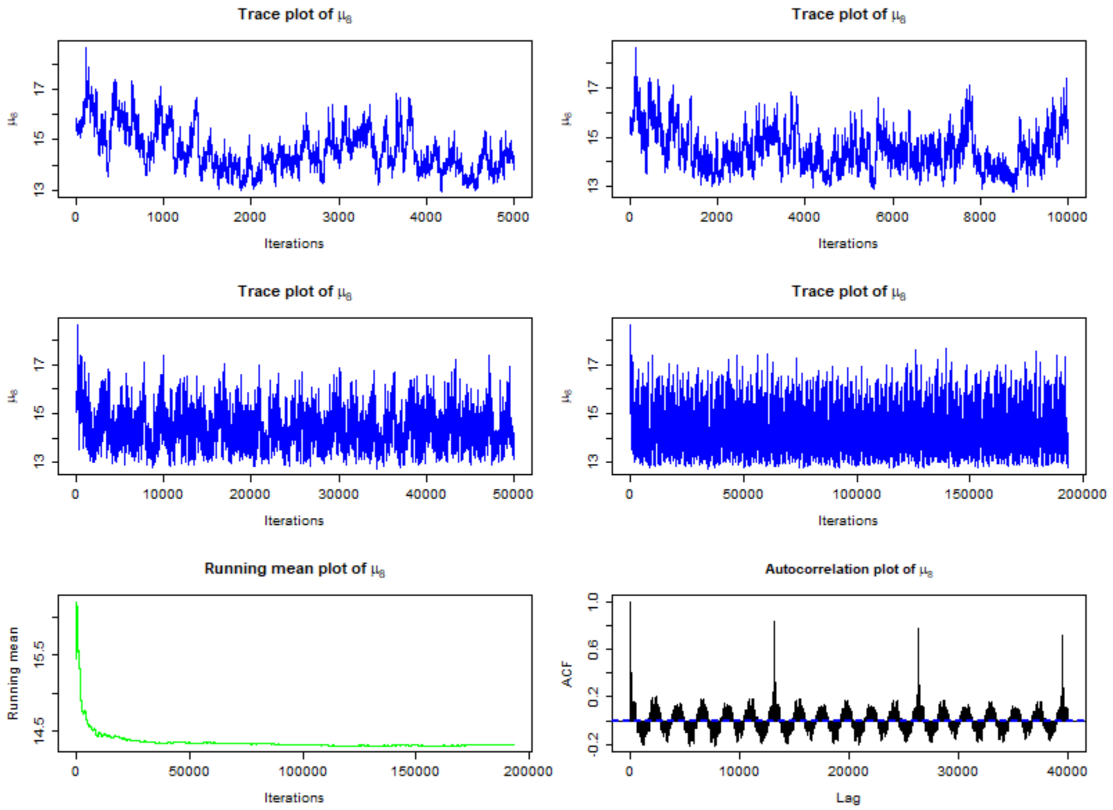

Figure 59: Plots for the parameter μ_8 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

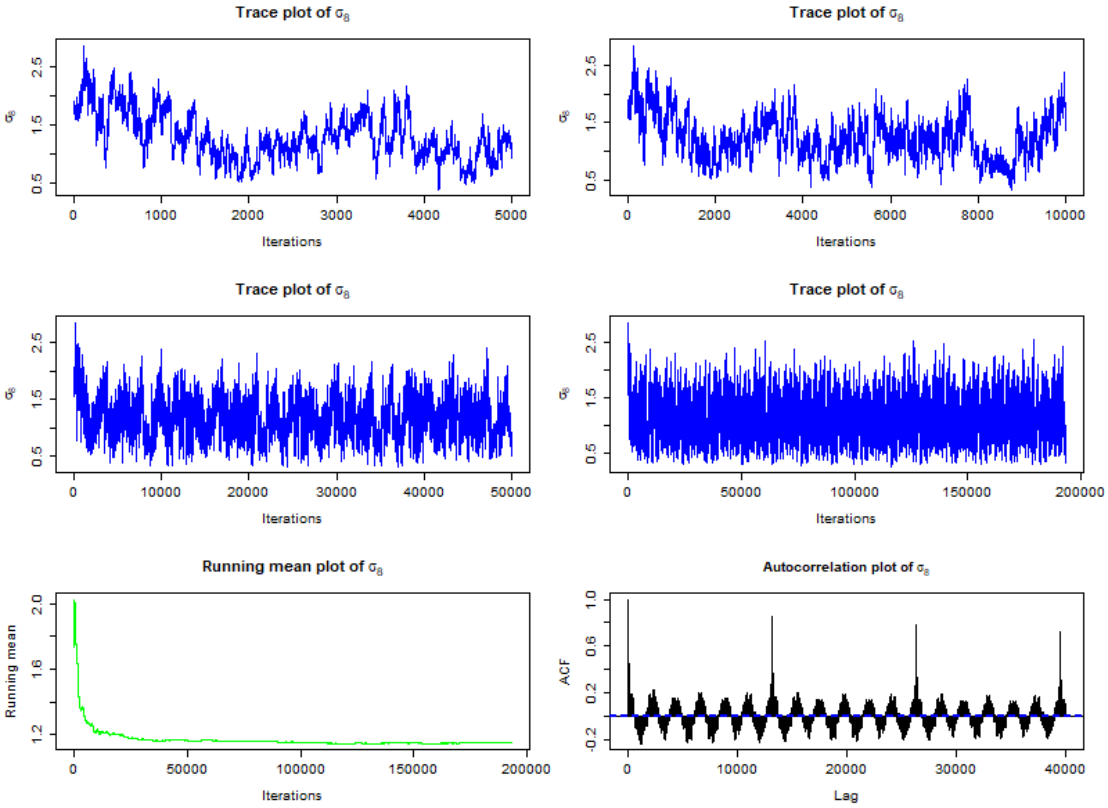

Figure 60: Plots for the parameter σ_8 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

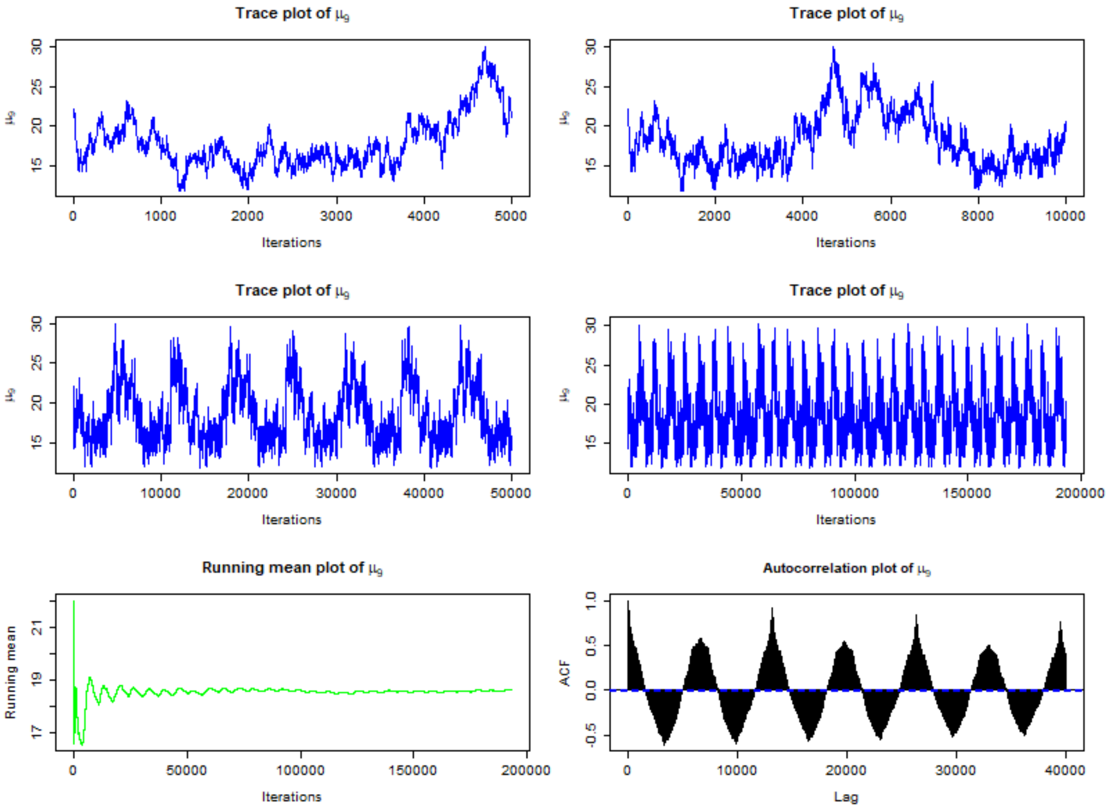

Figure 61: Plots for the parameter μ_9 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

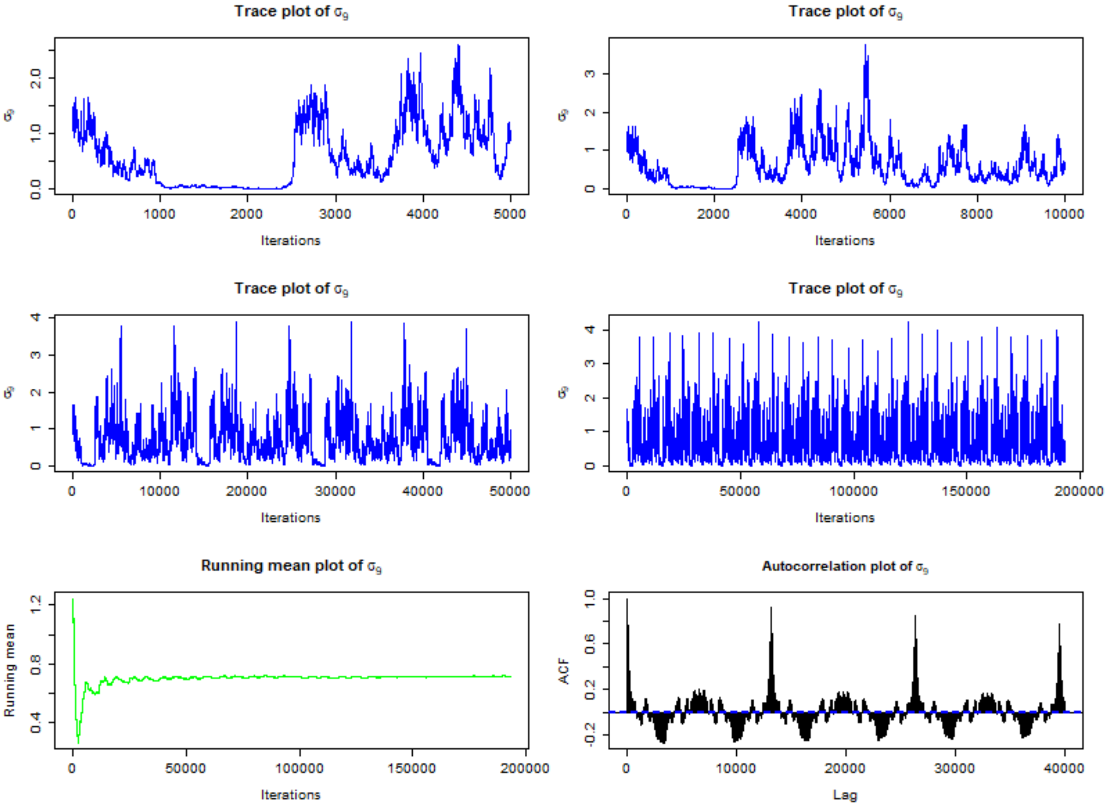

Figure 62: Plots for the parameter σ_9 based on the MCMC run for the line-wise model. The two upper rows show trace plots for the first 5000, 10000 and 50000 iterations, in addition to the trace plot for the full simulation. To the lower left the corresponding running mean plot, and to its right the autocorrelation plot.

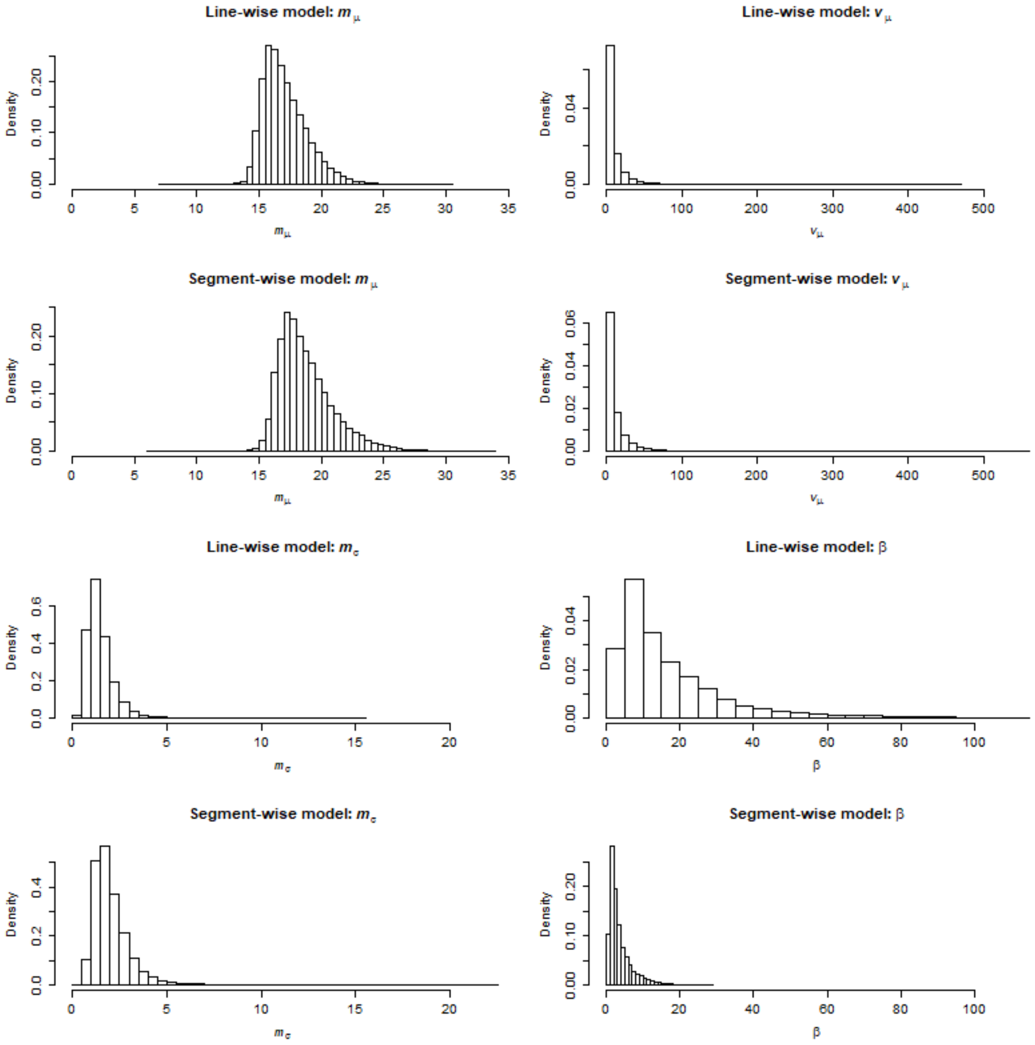

Figure 63: Histograms of the generated values for m_μ, v_μ, m_σ and β from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(m_\mu|x)$, $f(v_\mu|x)$, $f(m_\sigma|x)$ and $f(\beta|x)$.

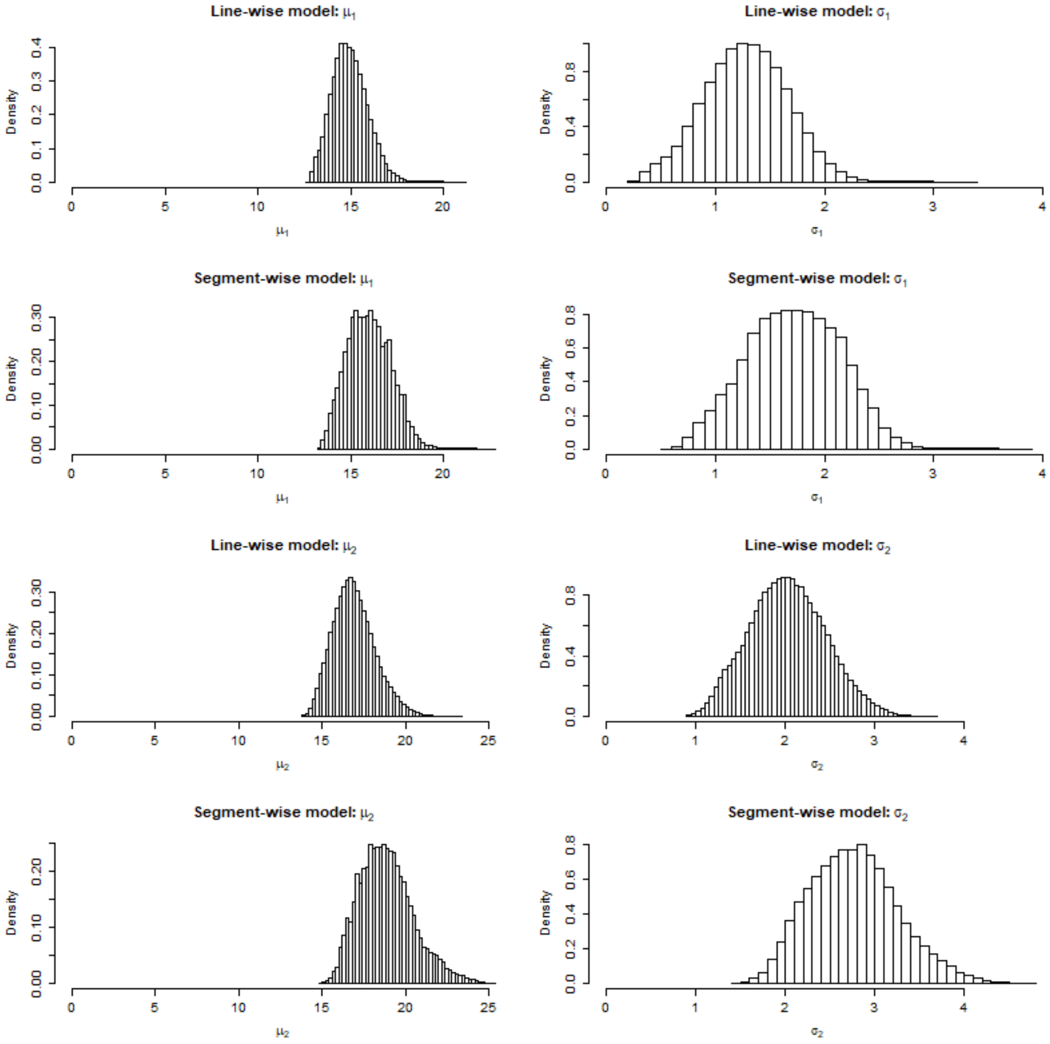

Figure 64: Histograms of the generated values for μ_1, σ_1, μ_2 and σ_2 from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(\mu_1|x)$, $f(\sigma_1|x)$, $f(\mu_2|x)$ and $f(\sigma_2|x)$.

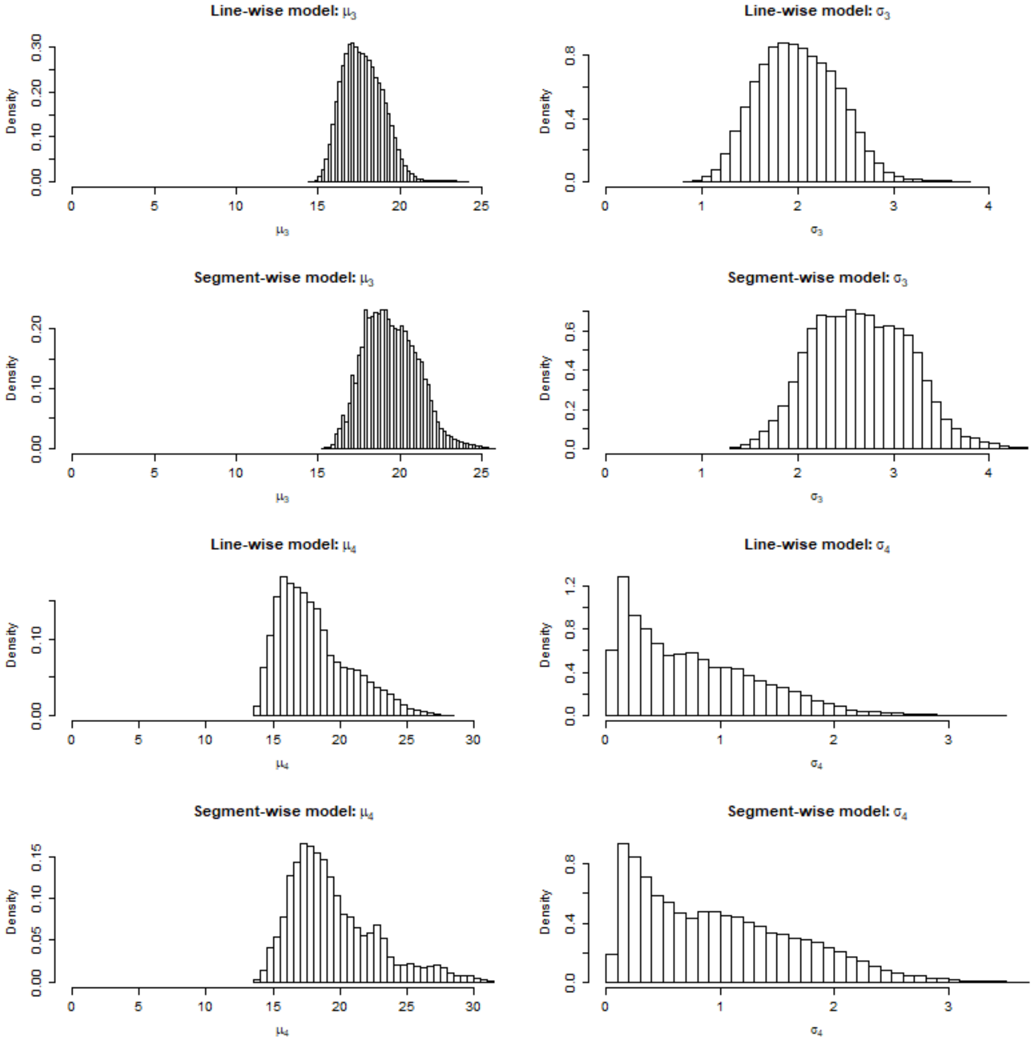

Figure 65: Histograms of the generated values for μ_3, σ_3, μ_4 and σ_4 from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(\mu_3|x)$, $f(\sigma_3|x)$, $f(\mu_4|x)$ and $f(\sigma_4|x)$.

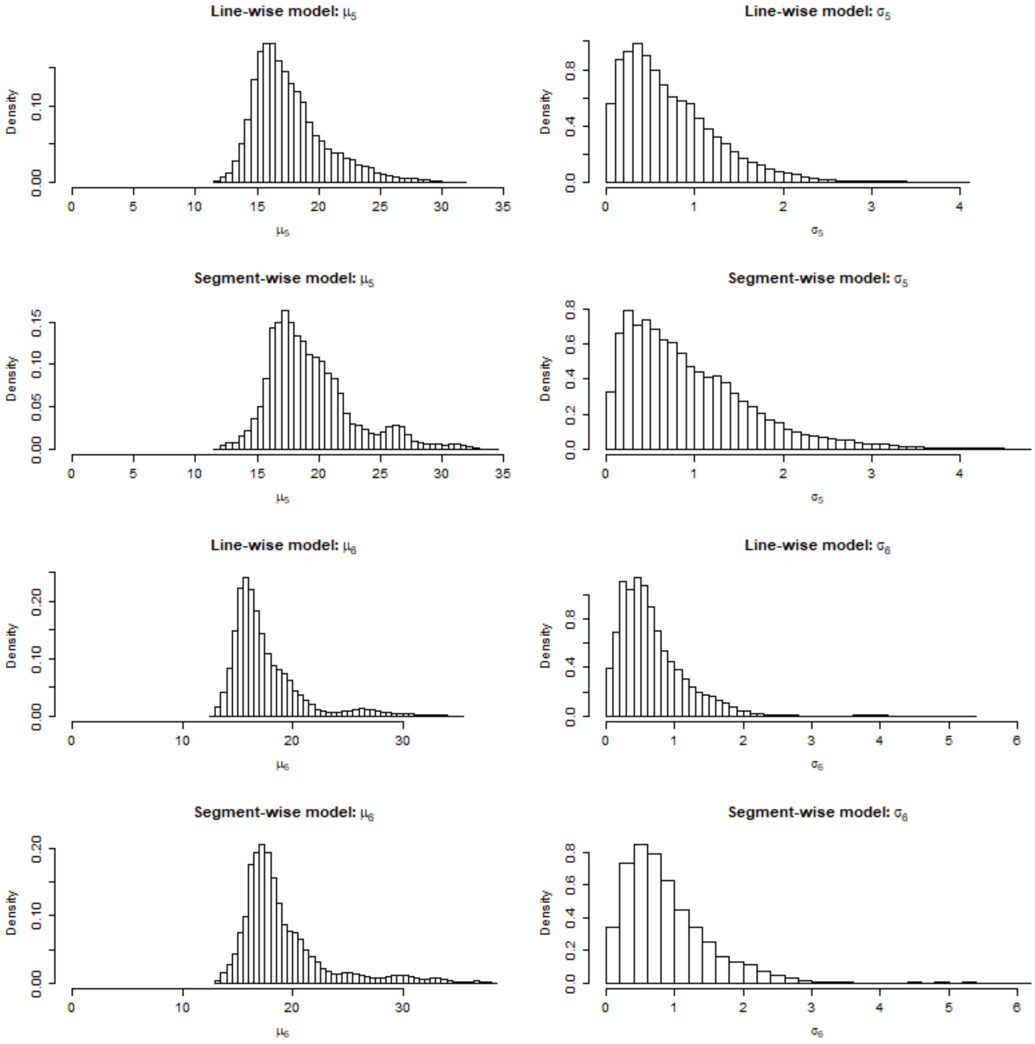

Figure 66: Histograms of the generated values for μ_5, σ_5, μ_6 and σ_6 from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(\mu_5|x)$, $f(\sigma_5|x)$, $f(\mu_6|x)$ and $f(\sigma_6|x)$.

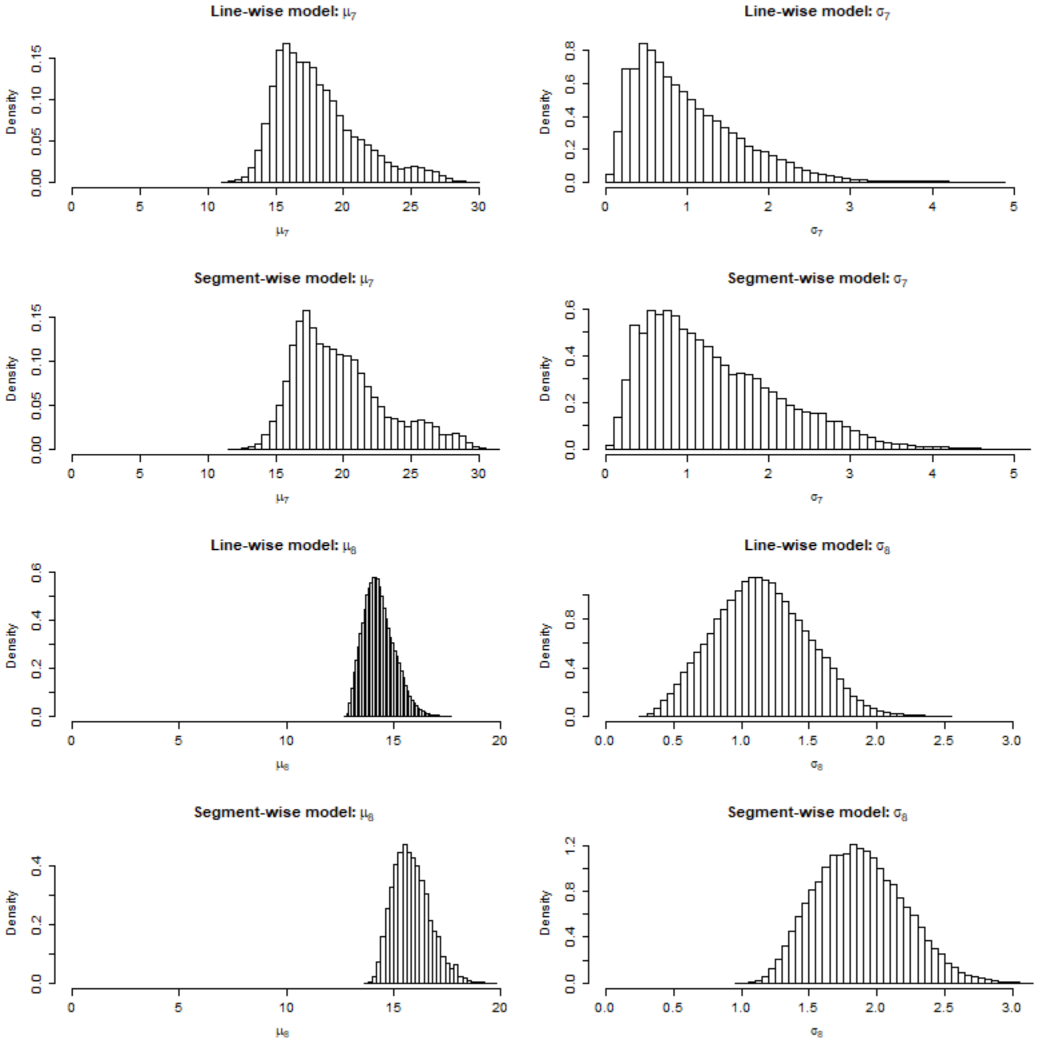

Figure 67: Histograms of the generated values for μ_7, σ_7, μ_8 and σ_8 from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(\mu_7|x)$, $f(\sigma_7|x)$, $f(\mu_8|x)$ and $f(\sigma_8|x)$.

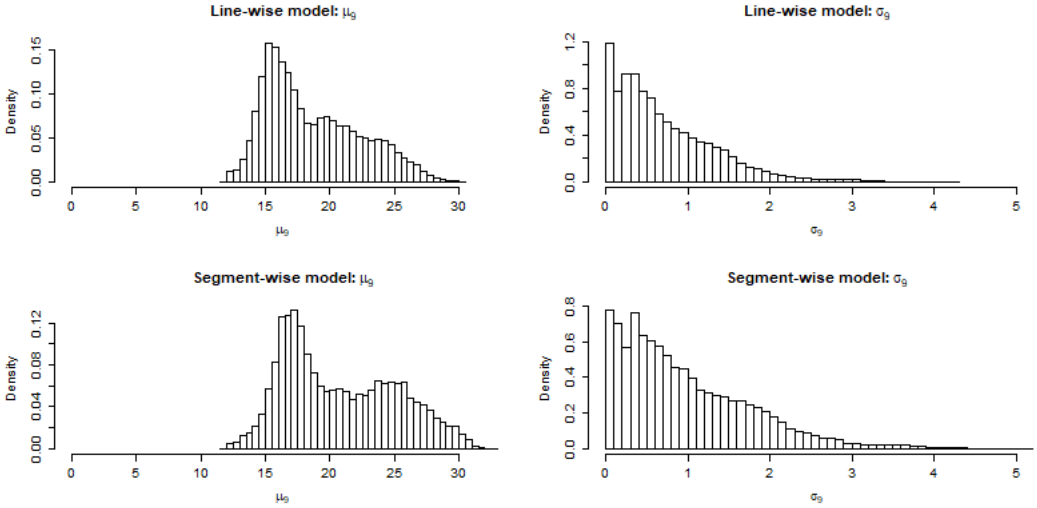

Figure 68: Histograms of the generated values for μ_9 and σ_9 from both the MCMC run based on the line-wise model and the segment-wise model. These values correspond to samples from the marginal posterior distributions $f(\mu_9|x)$ and $f(\sigma_9|x)$.

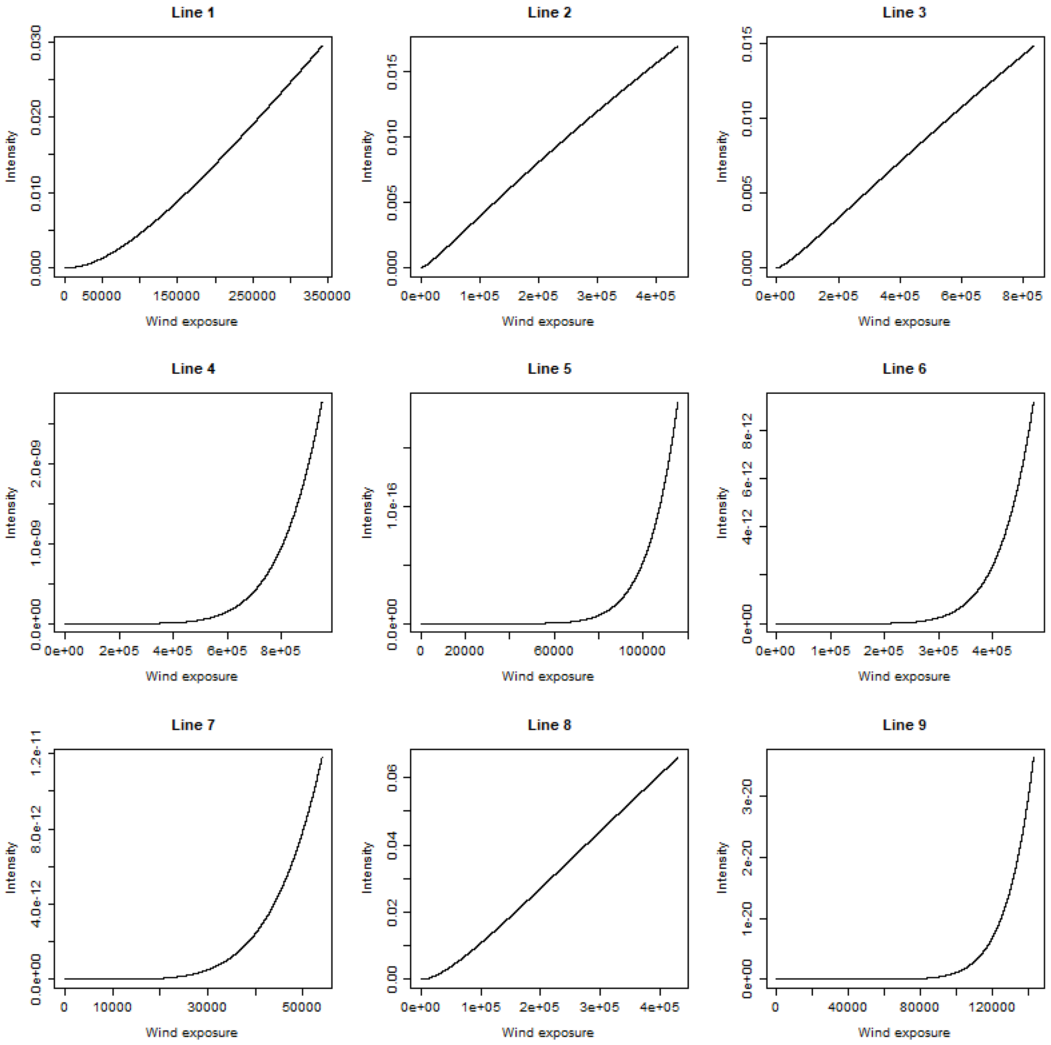

Figure 69: Intensity as a function of wind exposure for all lines based on the segment-wise model. The curves are based on the mean values of β, μ_l and σ_l from the corresponding MCMC run. The range of wind exposure is here slightly extended compared to for the case when $\beta = 1$, as is the case for the simulated data.

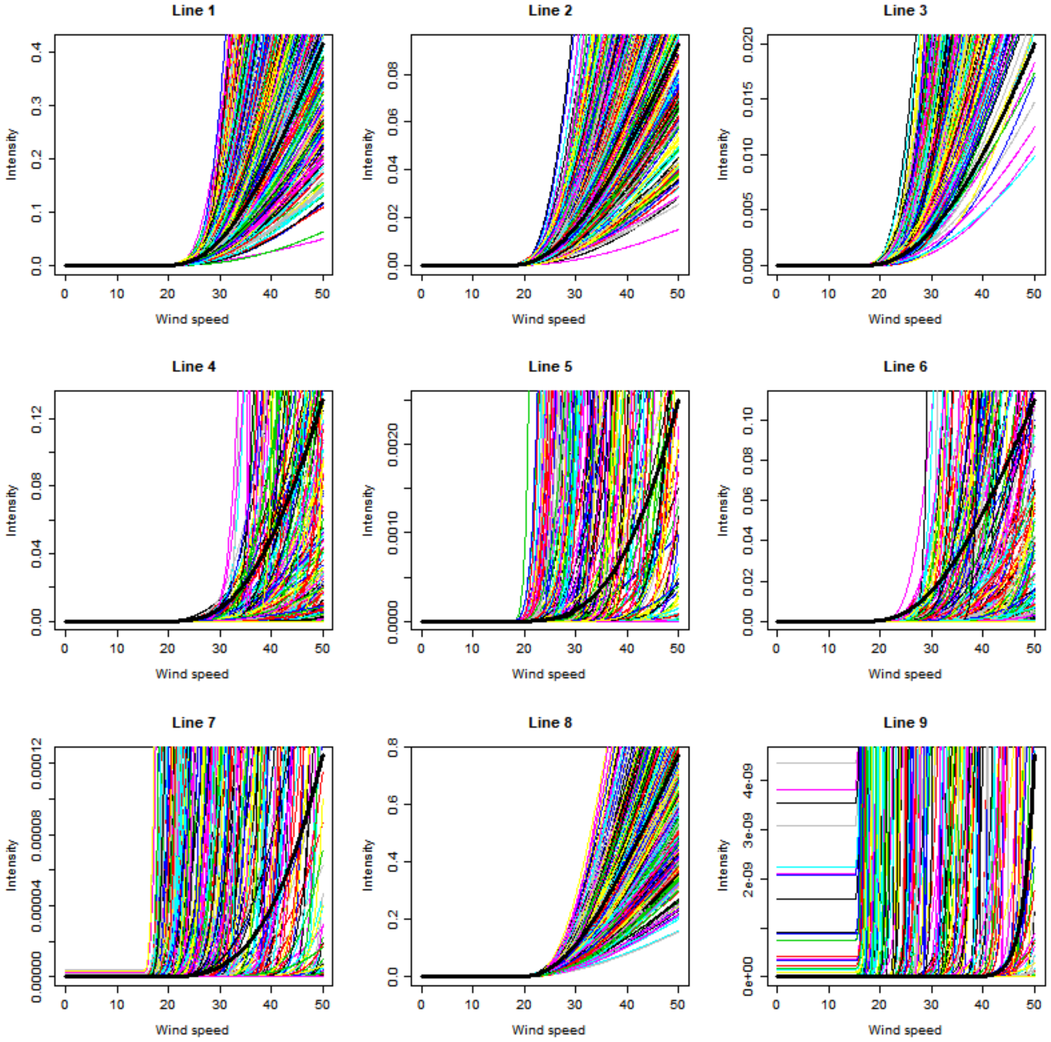

Figure 70: The intensity curve corresponding to every 100th iteration of the MCMC run for the segment-wise model. The thicker black line corresponds to the curve the paramters for the simulated data would give. Note that the y-axis differ from plot to plot.

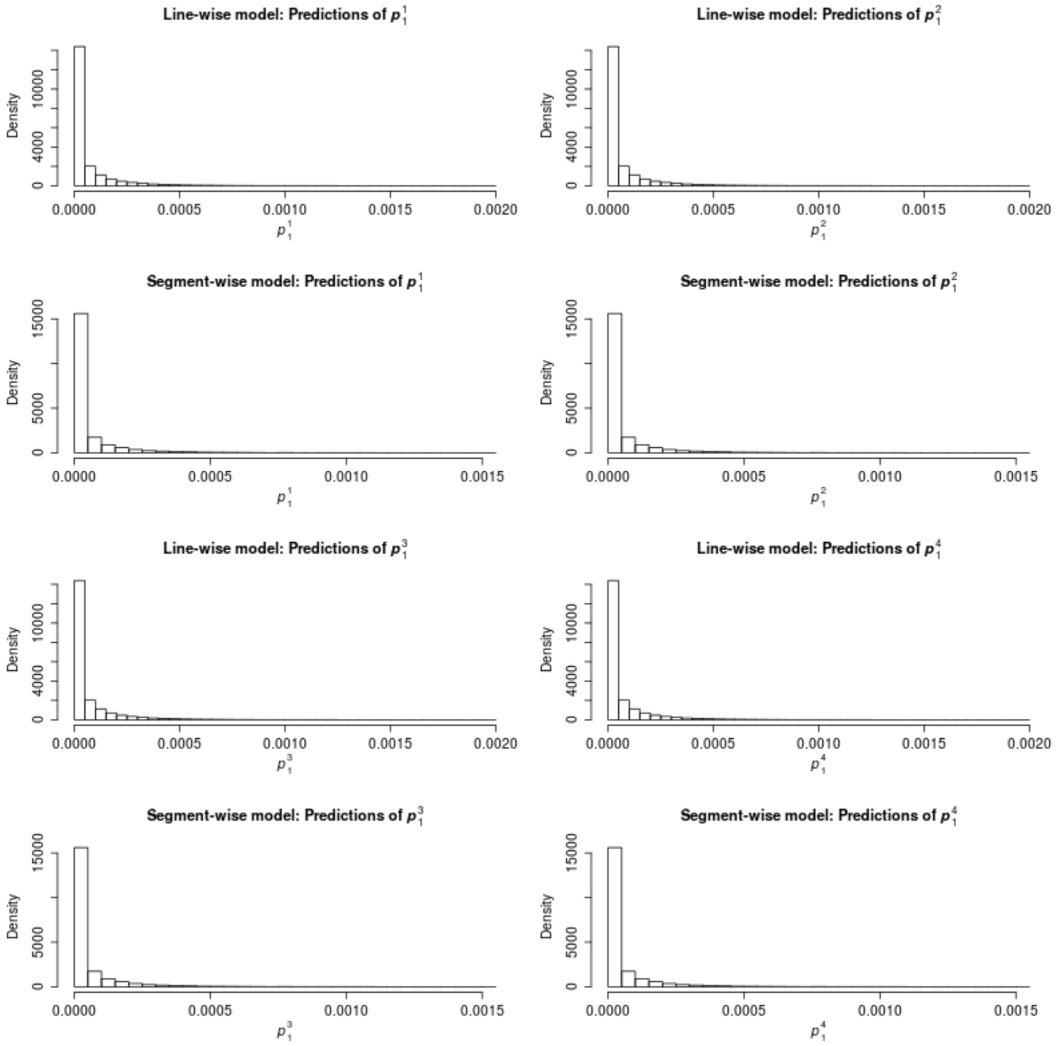

Figure 71: Predictions for the four first hours of July 1, 2014. The four next hours have the exact same predictions, and we therefore omit plotting them. Also the eight first hours of December 1, 2014, have the exact same plots.