Downloaded from orbit.dtu.dk on: May 07, 2024

DTU Library

Grey-box modelling and forecasting of stormwater flow in sewer systems

Bjerregård, Mathias Blicher

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Bjerregård, M. B. (2022). *Grey-box modelling and forecasting of stormwater flow in sewer systems*. Technical University of Denmark.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Grey-box modelling and forecasting of stormwater flow in sewer systems

Mathias Blicher Bjerregård

Technical University of Denmark

DTU

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Kongens Lyngby 2022

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

The goal of this thesis is to wrap up my endeavours in the realm of modelling and probabilistic forecasting. The work is anchored in my two quite different papers and is therefore essentially two-legged. The first leg deals with the development and estimation of a grey-box model suited for forecasting, and this is the primary subject of the thesis. In this case, I study the modelling and forecasting of rainfall-response in a Danish stormwater tunnel, but the focus is really on the modelling process and its inherent challenges rather than the specific case. This exploration would therefore be relevant for practical grey-box modellers within many fields, e.g. wind power forecasting and financial forecasting. The second leg deals with forecast evaluation, and is exemplified by applying some evaluation metrics to the stormwater forecasting model developed in the first leg followed by a discussion of what kind of value is gained from such an evaluation effort. Again, the relevance of the demonstrated work on forecast evaluation is not limited to the specific case, but can be equivalently applied to probabilistic forecasts in other areas.

Summary (Danish)

Målet med denne afhandling er at sammenfatte mine studier af modellering og fordelingsforudsigelser. Arbejdet er forankret i mine to ret forskellige artikler og kan derfor inddeles i to dele. Første del omhandler opsætning, udvikling og estimering af en grey-box model med henblik på forudsigelser, og dette er det primære emne i afhandlingen. I dette tilfælde studerer jeg modellering og forudsigelse af regnvejrsrespons i en dansk regnvandstunnel, men det egentlige fokus er på modelleringsprocessen og de udfordringer der opstår snarere end selve regnvejrsemnet. Dette studie kan derfor være relevant for anvendt grey-box modellering inden for mange områder, f.eks. vindenergi og finans. Anden del omhandler evaluering af forudigelser, som her bliver behandlet ved at anvende nogle evalueringsmetrikker på forudsigelserne fra regnvejrsmodellen fra første del. En diskussion af hvad sådan en evaluering af forudsigelserne fortæller, og hvilken værdi det giver følger naturligt. Igen er relevansen af den demonstrerede evaluering ikke blot begrænset til regnvandsforudsigelser, men kan anvendes tilsvarende på fordelingsforudsigelser inden for andre områder.

Preface

In 2015, I attended the Time Series Analysis course on DTU as a part of my Master's programme. This would be my first encounter with the topic of forecasting of real processes, anchored in rigorous statistical theory. The course would expose the participants to everything relevant for this topic, including parameter estimation, model selection, filtering as well as computation of ℓ-step predictions and associated covariances. Only linear methods and models were considered, but the content strongly piqued my interest, so naturally I continued to pursue this track by attending the Advanced Time Series Analysis course in the autumn of 2016. In stark contrast to the limited, safe, well-understood, linear methods from the ordinary Time Series Analysis, the advanced course was like the wild west. A chaotic potpourri of new and old, in some cases poorly documented, methods of high complexity. A seemingly unlimited collection of possible model classes to choose from. At first, the content felt overwhelming, but I soon understood that this was a pretty close representation of the practical reality of real world modelling and forecasting: the real world is generally non-linear and erratic, where each individual problem requires a tailor-made solution. Among the many models encountered in the course was the stochastic differential equation. I did not put much effort into studying it at the time, but it would later become the dominant model class in my studies.

In early 2017 I began to write my Master thesis. Because of my interest in forecasting, my supervisor suggested that I would work on the evaluation of forecasts, which I agreed to. After the Master's programme concluded, I continued to work on this topic with the intent of writing and publishing a review article with a focus on probabilistic forecast evaluation, in particular for multivariate forecasts. Despite the importance of the latter, there was no review available in the literature covering this family of forecasts, which thus made a great and very relevant case for my first paper.

My work on forecast evaluation soon transitioned into a PhD programme on forecasting and modelling of real, physical systems. This programme included a predefined case, which would feature modelling, forecasting and control of urban drainage systems with the study subject being a newly constructed stormwater tunnel in Singapore. Obviously, considerable amounts of energy is consumed in waste- and stormwater management, in particular due to the operation of pumps. The philosophy of the project was that a more optimized pump control strategy should be possible, given intelligent modelling and forecasting of the

stormwater management system. Furthermore, if the surrounding energy system could be integrated into the control scheme, the stormwater masses in the tunnel would effectively be a deferrable load, which could be pumped out of the system during the most cost-effective time window.

Stochastic differential equations appeared to constitute the ideal building blocks for a model capable of the above for three reasons. Firstly, they generate full probability distributions and hence complete information of the forecasts, which is necessary for optimal decision-making. Secondly, they can be integrated in model predictive control very naturally. Thirdly, when they have been estimated, new forecasts can be generated extremely fast when new data is obtained, which is a strong feature for control on a short time-scale.

Although the setup for the project was well in place, the modelling of the stormwater tunnel turned out to be a very challenging task, which consumed all the remaining time on my PhD programme. I am therefore tremendously happy that I have been able to finish and document a fully satisfactory model in the end. This PhD thesis summarizes the various challenges I faced on the way, and how I solved each and one of them.

Mathias Blicher Bjerregård, Lyngby 2022

Acknowledgements

I am incredibly grateful to everyone who has in any thinkable way contributed to make my time and work on the PhD programme enjoyable and constructive.

First of all, I of course want to thank my fantastic supervisors from the Section of Dynamical Systems, Lasse Engbo Christiansen and Jan Kloppenborg Møller for their patience and tireless commitment towards helping me succeed with my endeavours. I also want to thank my co-supervisors Morten Borup Hansen (DTU Environment) and Dusit Niyato (NTU Singapore) for their assisting supervision.

I feel that 'the Chieftain', as we sometimes call him in the office, who also goes by the name Henrik Madsen (prof. and head of the Section) has to be addressed in his own paragraph. Henrik has supported me and shown so much belief in my skill and potential all the way from the beginning until the conclusion, even when things looked really grave seen from my perspective. The value of having an engaging leader that is fully on the side of his employees and cares wholeheartedly about their well-being can not be overestimated.

Several other current or former staff members have been helpful to me and deserve my thanks, including Peter Steen Mikkelsen (DTU Environment), Kim Knudsen, Anette Iversen, Hanne Marie Jensen, Uffe Høgsbro Thygesen, Peder Bacher and Karsten Schmidt.

I have been fortunate to have so many wonderful colleagues at DTU, that I am afraid to forget someone, should I attempt to list them all. My office mates, Niclas Brabrand Brok, Maksim Mazuryn and Amos Schledorn are however unavoidable, and are undoubtedly the very best that the European engineering soup has to offer, both in terms of professional and friendly qualities. The meme culture has been consistently strong in office 303B/019, which has been very helpful for survival in the PhD circus. I also need to mention one of the wisest men I have met to date, my dear friend and colleague Razgar Ebrahimy, who I thank for all our casual, but very motivating chats, as well as for being a personal mentor for me in the last part of the programme.

Furthermore, Rocco Palmitessa from DTU Environment whom I went on external stay in Singapore simultaneously with has been a very friendly and competent colleague, who has helped me understand and find interest in urban drainage, the topic of my case study. My external stay in Singapore was a very

memorable experience, and I appreciate having some shared memories with Rocco on that account.

I am especially grateful to Lone Bo Jørgensen from HOFOR for her facilitation of and assistance with the Damhus tunnel data extraction, thus being vital in making that case study possible.

Naturally, I am very grateful to my beloved family for their support throughout the process. I am also extremely humbled that so many friends from far and beyond have followed my journey from the sideline, and shown all kinds of interest in my work as well as my well-being. In this context I want to mention Mikki Seidenschnur, Ole Lim Christiansen, Regitza Camilla Hansen, Alexander Askøe Olsen, Nis Christian Gellert, Christian Ankerstjerne Thilker, Frederik W. Andersen, Tobias Strand, Jonas Warming, Asbjørn Schack, Silas Sverre Christiansen, Thomas Meldgaard, Mette Skotte, Bolette Duun-Christensen, Line Andresen, James Hou, Hamidreza Moazzami, David Manstrup, Lisa Flam, Ulrik Damm, Alexander Licht, Jacob Simonsen and Martin Engberg.

Finally, the most special of my thanks goes to associated professor Per Bækgaard from the Section of Cognitive Systems, who has effectively acted as my rogue supervisor through a long series of top-secret coffee meetings. There is no limit to the passion, experience and insight of this man. Besides the PhD-related matters, our long sessions dedicated to important topics like birdwatching, Linux and radio technology have indeed added extra flavor to my PhD-study experience in the sense that I have been continuously reminded that there is a fantastic scientific world out there, only waiting for me to explore and immersive myself in.

List of publications

1. **Paper A**: Probabilistic forecasting of rainfall response in a Danish stormwater tunnel (*peer-reviewed, accepted and published in Journal of Hydrology*)
2. **Paper B**: An introduction to multivariate, probabilistic forecast evaluation (*peer-reviewed, accepted and published in Energy and AI*)

Contents

Chapter 1

Introduction

1.1 The importance of forecasting

The ability to forecast what is going to happen in the future is an art with an enormous impact on a virtually infinite range of problems. The reason why it is so impactful is because forecasts provide the end users with information, that allows them to *act* in the direction of their best interests (Hong et al., 2020). A rather innocent, but very relatable example, is when short-term weather forecasts are used by people for decision-making on their outdoor activities.

However, the impact of forecasting also applies to more serious problem types with higher stakes. In the financial sector, market forecasts are issued and exploited by traders constantly, in order to maximize profits and minimize losses (Xing et al., 2018). In the energy sector, wind power forecasts are very important because the power generation changes rapidly with changes in wind speed. Since power is sold on the day-ahead market before it is produced, too wrong wind power forecasts can end up costing the suppliers a lot of money (Costa et al., 2008).

These examples clearly demonstrate that knowing *what* object to forecast, *why* it is relevant and *who* is going to use it, are necessary in order to generate any value in a forecasting problem. But in order to take the problem to the practical level, it is also necessary to ask *how* to generate said forecasts, i.e. which method is going to produce them. This thesis specifies a forecasting problem according to these questions and then focuses on developing a model that can be used to produce the desired forecasts.

1.2 Forecasting of stormwater flow in sewer systems

The management of stormwater in urban areas makes a great case for a relevant forecasting problem.

Typical sewer systems are built primarily for wastewater drainage and can not handle too large amounts of stormwater in succession of rainfall events. Without a solution, heavy rainfall events will cause floods and inflict expensive damage on the affected urban area (Borup, 2014). A working solution is to implement some extra storage capacity that can contain the excess amounts of water, until it can be dealt with. This has been done in practice e.g. with the Damhus tunnel in a neighbourhood of Copenhagen in Denmark (Palmitessa et al., 2021).

Obviously, the accumulated stormwater from a rainfall event eventually needs to be removed from the extended storage capacity, such that new rainfall events can be handled. In the two real examples mentioned above, the stormwater ends up deep underground and has to be pumped out and led to a wastewater treatment plant, where it is cleansed before it is released into the adjacent sea.

The associated pumping activity consumes considerable amounts of energy due to the vast volumes of stormwater which are lifted from various depths. In the Damhus case this is currently handled by an automatic real-time control scheme that acts based on the current state of the system only, not on any forecasts. However, a more optimal strategy would be to apply model predictive control (MPC) (Brok et al., 2018). A smart MPC scheme could take advantage of the timely variations in the energy price, and schedule pumping for times where the energy prices are lowest. For this to work, it would be necessary to ensure that the hard constraint of not causing a flood is honored. Hence, reliable forecasts would be required. The end users in this case would be the local authorities who are financing the stormwater management (Lund et al., 2018).

Having identified stormwater forecasting as an interesting and relevant case study because of the potentially money-saving intelligent control prospects (the *what*, *why* and *who*), the most interesting question of *how* can then be addressed.

1.2.1 An overview of the state-of-the-art in rainfall-runoff modelling

For the already mentioned reasons, stormwater forecasting is already applied in the industry. In the following, a brief summary of the state-of-the-art for rainfall-runoff modelling is provided, inspired by Sitterson et al. (2018) and Jehanzaib et al. (2022). Rainfall-runoff models can be categorized into physical, empirical and conceptual models.

Physical models are deterministic white-box models of very high complexity. They are always highly accurate but suffer from comparatively slow computational speed. Some examples of physical models are the MIKE Urban (DHI, 2019) and the Storm Water Management Model (SWMM) (Rossman et al., 2010).

Conversely, *empirical* models are black-box models with no physical interpretation. They are purely data-driven, which has the advantages that no prior knowledge about the modelling case at hand is needed for model estimation, as well as having a low number of parameters to estimate. A major drawback is the coverage of data needed for reliable forecasting. If new events of a kind which are not reflected by the data used to train an empirical model occur, then the model cannot be expected to perform well. Some examples of empirical models are SCS-Curve Number models (Mishra et al., 2003) and machine learning methods like artificial neural networks (Yokoo et al., 2022).

Conceptual models are grey-box models that can be regarded as a compromise between physical and empirical models. They draw on the most crucial physical concepts without having to model every little corner of the catchment and every single pipe and manhole of the target case. Whatever is not captured by the physical parts of the conceptual model is accounted for by adding some appropriate uncertainty structure. Some examples of conceptual models are the TOPMODEL (Beven and Kirkby, 1979), the Hydrologiska Byråns Vattenbalansavdelning (HBV) (Dakhlaoui et al., 2012), the Hydrological Simulation Program-Fortran (HSPF) (Mohamoud and Prieto, 2012), and last but not the least, the continuous-discrete-time stochastic state-space model (CTSM) (Breinholt et al., 2011; Juhl et al., 2013), which is the model used in this thesis.

Furthermore, rainfall-runoff models are also distinguished based on spatial variability in the catchment. The relevant categories are fully distributed, semi-distributed and lumped models. Lumped models do not consider spatial variability and are thus the simplest. On the other hand, fully distributed models ideally feature a full description of the spatial variability, i.e. the rainfall-runoff hydrograph behaves differently in every single square meter of the cathcment.

The semi-distributed model is a compromise between the two, where the catchment is divided into a number of sub-catchments. Each sub-catchment behaves like a lumped model locally, but has its own unique parameter estimates which ensures some spatial variability on the global level (Sitterson et al., 2018). The CTSM used in this thesis is a lumped model, but could in theory be upgraded to a semi-distributed model if deemed necessary.

A look into recent applications of rainfall-runoff modelling reveals that all of the model categories are still being used and developed. Machine learning methods are both used in empirical models, see e.g. Van et al. (2020) and Tikhamarine et al. (2020), but also for improving the output from conceptual models, thus yielding hybrids between the two (Okkan et al., 2021). Purely conceptual models are still researched for their computational speed (Lavtar et al., 2019; Lees et al., 2021; Nearing et al., 2020) and physical models like SWMM are often applied for validation, where accuracy is valued more than time (Perin et al., 2020; Sañudo et al., 2020). For the future, researchers generally recommend to pursue further development of the machine learning route because of its ability to produce accurate forecasts without much physical, geographical and infrastructural understanding of every individual case, which means it has great potential for generalization and hence ubiquitous applicability. However, in my opinion, the conceptual models continue to be extremely relevant due to their unique combination of both being physically interpretable while at the same time having a small number of parameters. Furthermore, they tend to be fast at both deterministic simulation and uncertainty forecasting (Breinholt et al., 2012) and hence relevant for practical applications involving control, optimization and risk management in urban drainage.

1.2.2 Grey-box modelling as a base for forecasting

The ambition of being able to implement a future MPC scheme on the Damhus case introduced in Section 1.2 relies on the ability to constantly adapt to a changing environment and generate new forecasts quickly. If the forecasts must be computed fast, then they should not be based on complex physical models or data-heavy empirical models which are generally still too slow (Löwe et al., 2022; Su et al., 2019; Zhao et al., 2019), although examples of MPC integrated with physical models like the SWMM have been published (Sun et al., 2020). Instead, it is appealing to base the forecasting on one of the simpler conceptual models. But with simplification follows increased uncertainty. Improper handling of uncertainty could quickly lead to financial losses or even floods, so an alternative forecasting model must also cover this aspect.

A *grey-box model* such as the continuous-discrete-time stochastic state-space

model (CTSM) (see Chapter 2 for details) belongs to the category of conceptual models and seems like a compelling choice satisfying all of the above requirements. Grey-box models are usually data-driven, allowing them to be built on extremely simplified physical descriptions. The simple structure of the CTSM allows for simulation and hence forecast generation at a high computational speed, and crucially, they have a stochastic part which ensures that any uncertainty can be properly modelled as well. Indeed, a few CTSMs for urban drainage modelling have been published (Breinholt et al., 2011, 2012; Löwe et al., 2014, 2016) which provide a good starting point for this particular case study. Aside from the referenced papers, suprisingly, it has not been possible to find any recent work on CTSMs for stormwater forecasting, most likely due to the hype around machine learning approaches which tends to affect every modelling field nowadays. As a final remark, it is already well-known that grey-box models are generally very suited for MPC schemes (Thilker et al., 2021). Therefore, this thesis aims to develop a grey-box model, specifically a CTSM, for stormwater forecasting in the Damhus drainage system.

1.3 Objectives

The main objective of this thesis is to provide a comprehensive report on my grey-box modelling of the stormwater response in a Danish sewer system, namely the sewer system associated with the Damhus catchment in Copenhagen. To spell that out wordly, it means I want to forecast how much water flows into a stormwater tunnel at any given time, whenever it rains in the neighbourhood.

Meeting this objective yields two levels of contribution. Firstly, the modelling case features *non-linear* stormwater response and hence demonstrates a way to deal with this common problem. The principles used for solving it are quite generic and can thus be expected to be applicable for other sewer systems too, making this case study relevant for future work in the field of urban drainage modelling.

Secondly, actually building the model structure, estimating its parameters, evaluate its performance and then iterate over this modelling framework until a satisfying model has been achieved has been a time-consuming, challenging piece of work. It has forced me to deal with a range of frustrating obstacles encountered on the way, which relevance are not necessarily restricted to urban drainage modelling only. I can easily imagine other grey-box modellers will encounter these types of problems again and again in many different applications. By exposing those problems and my solutions to them in this thesis, I believe I can help other people solve some of their modelling problems faster, or even avoid

them before they arise.

Finally, this thesis has a parallel objective of investigating methods for multivariate probabilistic forecast evaluation. This is primarily covered in Paper B on its own. However, the forecasts produced by the developed grey-box model are indeed both probabilistic and multivariate, and since the purpose of the model is to produce proper forecasts, it makes a perfect case for a demonstration of how the evaluation methods discussed in Paper B may be applied to a real forecasting problem. Hence, the last objective of the thesis is to demonstrate how to properly evaluate the multivariate probabilistic forecasts of stormwater issued by the developed grey-box model.

1.4 Outline of the thesis

The thesis is organized as follows:

- Chapter 2 recaps how grey-box models are built from stochastic differential equations and estimated using the maximum likelihood principle.
- Chapter 3 explains how stormwater flow can be modelled with linear reservoir models.
- Chapter 4 combines the above methods to develop a grey-box model for forecasting of stormwater response in the Damhus tunnel.
- Chapter 5 features a demonstration of how the forecasts generated by the developed grey-box model can be evaluated.
- Finally, some concluding remarks on the presented work is provided in Chapter 6.

As a reader's guide, I strongly recommend reading Chapter 2.1-2.2 and all of Chapter 3, as the most fundamental mathematical basis is covered there. Section 2.3 serves as a reference for parameter estimation and is optional.

Chapter 4 is structured such that Section 4.1 and 4.3 can be read without 4.2, intended for the reader that just wants to understand the Damhus case and the fully developed model. Section 4.2 on the other hand, is a comprehensive section containing the more interesting details seen from a modeller's perspective. It is thus intended for the deep readthrough.

Chapter 5 connects the two publications which this thesis is based on. It is pretty brief and I recommend reading all of it.

That sums up the introduction to this PhD thesis. Enjoy the ride!

Chapter 2

Grey-box modelling

This chapter introduces the grey-box modelling skeleton used throughout this study. The idea of a grey-box model is to describe both the deterministic as well as the stochastic behaviour of the target system. The deterministic part is typically derived from well-understood physics, while the stochastic part should cover the distribution of system noise which is not captured by the deterministic physics (Tulleken, 1993). Stochastic differential equations (SDEs) are introduced as the perfect building blocks for such grey-box models. Furthermore, it is explained how a set of SDEs can be assembled to form a continuous-discrete-time state-space model (CTSM), which enables reconstruction of both observed and unobserved states of the target system. Finally, the parameter estimation framework for the CTSM is briefly summarized.

2.1 Stochastic differential equations

In the context of grey-models, SDEs provide a realistic way to describe the evolution of random variables, which are affected by some physical drift, typically with respect to time. A basic SDE may be given on the form (Øksendal, 2003):

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(X_t, t)\mathrm{d}W_t. \tag{2.1}$$

Here, t is continuous time, X_t is a random variable, $f(X_t, t)$ is called the drift function and describes the physical behaviour of X_t, while $g(X_t, t)$ is called the diffusion function and describes the stochastic behaviour of X_t. W_t is a Wiener process, sometimes also referred to as a random walk or standard Brownian motion, i.e. it has the property that the 'steps' $W_{t+\mathrm{d}t} - W_t$ are independent and identically distributed (i.i.d) with $W_{t+\mathrm{d}t} - W_t \sim \mathcal{N}(0, \mathrm{d}t^2)$ (Wiener, 1923).

A very simple example of a SDE is for instance,

$$\mathrm{d}X_t = -X_t\mathrm{d}t + \sigma\mathrm{d}W_t, \tag{2.2}$$

where σ is a constant. The drift term $f(X_t, t)\mathrm{d}t = -X_t\mathrm{d}t$ drives the expectation $\mathrm{E}[X_t]$ towards 0 over time, while the diffusion term causes the uncertainty to tend towards a Gaussian distribution with variance $\mathrm{V}[X_t] = \sigma^2$. The evolution of the Eq. (2.2) is visualized in Fig. 2.1 with $X_0 = 1$ and $\sigma = 0.05$. Because the evolution of W_t is random, the path of X_t will be different every time it is realized. If X_t is realized over and over again, an ensemble of realizations is obtained which in turn constitutes an approximation to the full probabilistic evolution of the SDE. From this ensemble, any desired probabilistic structure can be extracted, such as (univariate) marginal forecast densities, (multivariate) joint forecast densities, quantiles or expectations. This is also demonstrated in Fig. 2.1, by highlighting the marginal forecast density at time $t = 2$.

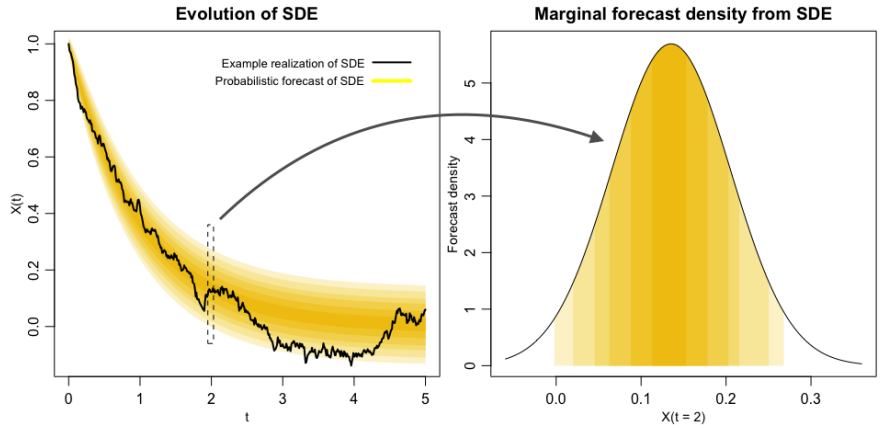

Figure 2.1: Evolution of the the SDE in Eq. (2.2) with $X_0 = 1$ and $\sigma = 0.05$. The marginal forecast density at time $t = 2$ is shown to the right. The yellow bands cover up to and including the 95%-prediction interval in both plots.

2.2 The continuous-discrete-time state-space model

It is now established that SDEs are useful for describing the behaviour of real systems. In practice, it is usually necessary to formulate a number of inter-dependent SDEs to mimic the dynamics of the target system reasonably. The next step is to determine the parameter values. Of course, parameter values may simply be guessed or estimated offline. However, in grey-box modelling, a data-driven approach is typically taken by observing the system and use the obtained data to estimate the most likely parameter values.

A common way to realize this idea is to formulate a continuous-discrete-time

state-space model (CTSM):

$$\begin{aligned} \mathrm{d}X_t &= f(X_t, U_t, t)\mathrm{d}t + g(X_t, U_t, t)\mathrm{d}W_t, \\ Y_k &= h(X_{t_k}, U_{t_k}, t_k) + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2). \end{aligned} \tag{2.3}$$

The first equation is called the system equation and governs the system behaviour in continuous time. It is a generalized version of Eq. (2.1) in the sense that X_t is now a vector of a number of system states, and both the drift and diffusion terms are also functions of external inputs, U_t in addition to states and time. The second equation is called the observation equation which models the observation as a random variable, Y_k. Its core interpretation is that the continuous-time system is only observed at discrete time points, t_k, hence giving rise to the tag 'continuous-discrete-time'. Generally, Y_k is a vector, but since this thesis exclusively features CTSMs with only one observed state, Y_k will from now on be regarded as a univariate random variable.

An example of a CTSM with two states X_1 and X_2, where only the latter is observed is given below,

$$\begin{aligned} \mathrm{d}X_{1,t} &= -X_{1,t}\mathrm{d}t + \sigma\mathrm{d}W_{1,t} \\ \mathrm{d}X_{2,t} &= X_{1,t}\mathrm{d}t + \sigma\mathrm{d}W_{2,t} \\ Y_k &= X_{2,t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2), \end{aligned} \tag{2.4}$$

which can be rewritten into the form in Eq. (2.3), by putting $X_t = (X_{1,t}, X_{2,t})'$ and $W_t = (W_{1,t}, W_{2,t})'$:

$$\begin{aligned} \mathrm{d}X_t &= \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} X_t\mathrm{d}t + \sigma\mathrm{d}W_t, \\ Y_k &= \begin{pmatrix} 0 & 1 \end{pmatrix} X_{t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2). \end{aligned} \tag{2.5}$$

Due to being unobserved, a state like $X_{1,t}$ is popularly called a *hidden* state, but crucially it can still be reconstructed due to its interaction with the observed state, in this case $X_{2,t}$. This is the most important feature of the CTSM - every relevant state of the system is subject to reconstruction, even though only a part of the system is directly observed.

2.3 Model estimation

When the exact structure of the CTSM has been chosen, the final step to deliver a fully functional model is the estimation of its parameters. For grey-box models it is a very common practice to base parameter estimation on the maximum

likelihood principle (Kristensen et al., 2004) and the extended Kalman filter (Hoshiya and Saito, 1984). The theory of this framework is outlined below, while in practice it is all handled automatically by the R-package `ctsmr` (Juhl et al., 2013).

2.3.1 The likelihood principle

Let θ be the vector of parameters for the grey-box model subject to estimation, let $\mathcal{Y} = (y_0, \cdots, y_N)$ be a series of observations of the modelled system, and let $\phi(\mathcal{Y}|\theta)$ be the joint probability of observing $y_0, \cdots, y_N$ given θ. Then the likelihood of the model, $L(\theta|\mathcal{Y})$ is simply

$$L(\theta|\mathcal{Y}) = \phi(\mathcal{Y}|\theta). \tag{2.6}$$

The goal is to identify the set of parameters $\hat{\theta}$ that maximizes the likelihood, i.e. maximizes the joint probability of having observed exactly $\mathcal{Y}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ L(\theta|\mathcal{Y}). \tag{2.7}$$

The joint probability can be expanded into a chain of conditional probabilities, using the general product rule $P(A \cap B) = P(A|B)(P(B)$ (Pitman, 1999),

$$\begin{aligned} \phi(\mathcal{Y}|\theta) &= \phi(y_0, \cdots, y_N|\theta) \\ &= \phi(y_N|y_{N-1}, \cdots, y_0, \theta) \cdots \phi(y_1|y_0, \theta) \cdot \phi(y_0|\theta). \end{aligned} \tag{2.8}$$

Since every factor in Eq. (2.8) is a probability and hence a number between 0 and 1, this product quickly becomes an extremely small number. In most modelling cases there will be hundreds or thousands of observations, and then the product becomes so small that it is computationally impractical to work with. Instead, consider the logarithm of the probability,

$$\begin{aligned} \log(\phi(\mathcal{Y}|\theta)) = \log(\phi(y_N|y_{N-1}, \cdots, y_0, \theta)) + ... \\ + \log(\phi(y_1|y_0, \theta)) + \log(\phi(y_0|\theta)). \end{aligned} \tag{2.9}$$

This is a sum and is hence computationally stable for practically infinitely large sets of observations. Taking the logarithm on both sides of Eq. (2.6) gives

$$\log(L(\theta|\mathcal{Y})) = \log(\phi(\mathcal{Y}|\theta)), \tag{2.10}$$

where the left-hand side is called the *log*-likelihood. Fortunately the maximum likelihood estimate is invariant under this log transformation, and thus,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ \log(L(\theta|\mathcal{Y})). \tag{2.11}$$

Finally, it is common practice to think of the likelihood as a loss that has to be minimized. For this reason, the *negative* log-likelihood,

$$\ell(\theta|\mathcal{Y}) = -\log(L(\theta|\mathcal{Y})), \tag{2.12}$$

is often used instead. It follows that

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ \ell(\theta|\mathcal{Y}). \tag{2.13}$$

Fundamentally it does not really matter if the negative form is used or not, and the method is still called "maximum likelihood" estimation even though Eq. (2.13) rather looks like a "minimum negative log-likelihood" estimation. Either way, throughout this thesis the negative form, $\ell(\theta|\mathcal{Y})$, is used.

Clearly, in order to be able to calculate $\ell(\theta|\mathcal{Y})$, a way to calculate the conditional probabilities listed in Eq. (2.9) is needed. This is handled by the extended Kalman filter, as outlined in the next subsection.

2.3.2 Kalman filtering

Consider again a CTSM as formulated in Eq. (2.3). In short, the Kalman filter is an algorithm that alternates between doing one-step *prediction* and *reconstruction*, respectively, of both the state expectation and covariance, where 'one-step' refers to moving in time from t_k to t_{k+1}. The predictions are needed to generate input for the negative log-likelihood. The reconstructions provide informed estimates of both observed and hidden states, see Fig. 2.2.

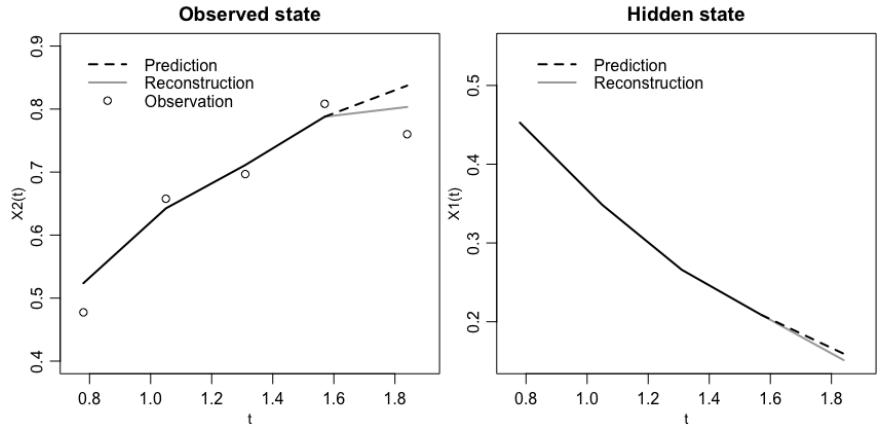

Figure 2.2: Example of the usage of a Kalman filter for state reconstruction on the system in Eq. (2.4) with $\sigma = 0.05$ and $\sigma_\varepsilon = 0.05$.

In the following, a very brief introduction to the Kalman filtering principle is outlined, with most mathematical details omitted. The sole intent is to give the reader an idea of how the filtering ultimately enables parameter estimation through the negative log-likelihood.

The state prediction $\hat{x}_{k+1|k}$ is found by letting the system equation propagate forward in time. While the state is regarded as a continuous-time process, in practice it is evaluated in very small discrete steps Δt, using the Euler-Maruyama method:

$$X_{t+\Delta t} = X_t + f(X_t, U_t, t)\Delta t + g(X_t, U_t, t)(W_{t+\Delta t} - W_t). \tag{2.14}$$

This recursion can be continued from $t = t_k$ up to $t = t_{k+1}$, where the next observation is due to be made, and hence:

$$\hat{x}_{k+1|k} = X_{t_{k+1}|t_k}. \tag{2.15}$$

The predicted observation, $\hat{y}_{k+1|k}$, is then derived from Eq. (2.3),

$$\hat{y}_{k+1|k} = h(X_{t_{k+1}|t_k}, U_{t_{k+1}}, t_{k+1}) \tag{2.16}$$

as is the predicted observation noise $\hat{\sigma}_{k+1|k}$ in a similar manner.

As soon as the new observation y_{k+1} is available, the reconstruction step can be executed. The reconstructed state $\hat{x}_{k+1|k+1}$ is calculated as

$$\hat{x}_{k+1|k+1} = \mathrm{E}[X_{t_{k+1}}|y_{k+1}], \tag{2.17}$$

with further elaboration omitted. The same goes for the predicted and reconstructed state covariance, see e.g. Madsen (2007) for more details. Then, $X_{t_k+1} = \hat{x}_{k+1|k+1}$ can be used as initial condition for the next prediction step from t_{k+1} to t_{k+2}. Continuing this alternating prediction/reconstruction algorithm results in a series of one-step predictions.

Now, recall that the goal is to calculate $\ell(\theta|\mathcal{Y})$, for which the conditional probabilities in Eq. (2.9) are required. Assume for a moment that each conditional probability $\phi(y_{k+1}|y_k, \cdots, y_0, \theta)$ has the Markov property and can be characterized by a Gaussian distribution with mean $\hat{y}_{k+1|k}$ and variance $\hat{\sigma}^2_{k+1|k}$:

$$Y_{k+1|k,\cdots,0,\theta} \sim \mathcal{N}(\hat{y}_{k+1|k}, \hat{\sigma}^2_{k+1|k}). \tag{2.18}$$

Assume also that all of the conditional probabilities are independent. Then the negative log-likelihood can be derived from the multivariate log-normal distribution and becomes,

$$\ell(\theta|\mathcal{Y}) = \frac{N}{2}\log(2\pi) + \frac{1}{2}\sum_{k=0}^{N-1}\left(\log(\hat{\sigma}^2_{k+1|k}) + \left(\frac{y_{k+1} - \hat{y}_{k+1|k}}{\hat{\sigma}_{k+1|k}}\right)^2\right). \tag{2.19}$$

The assumptions above are in fact true, if and only if the CTSM is linear. However, real systems are rarely linear in practical applications, so the non-linear case must also be considered. In the non-linear case, the CTSM (Eq. (2.3)) is linearized at each time step prior to prediction and reconstruction. This modification is what defines the *extended* Kalman filter (Hoshiya and Saito, 1984).

Fortunately, it turns out that if the time increments Δt in the evolution of a non-linear system are sufficiently small, then the characterization in Eq. (2.18) is asymptotically approached. Consequently, it is viable to use the one-step predictions of a non-linear CTSM provided by the extended Kalman filter, for calculation of the negative log-likelihood (Brok et al., 2018).

In order to estimate the parameters of the CTSM, the negative log-likelihood then has to be minimized w.r.t θ, and then this entire routine is passed to an optimizing algorithm of choice. Throughout this thesis, the R-routine `nlminb` is used for optimization. It is based on an L-BFGS-B method (Zhu et al., 1997). Further details on optimization are not included in this thesis. For a pseudo-code summary of the model estimation framework, see the Supplementary material of Paper A (section E).

Chapter 3

Modelling stormwater flow

Chapter 2 recapped the CTSM-framework for the modelling of dynamical systems in general. Paper A, however, focuses on a specific kind of dynamical system, namely, rainfall-response in a stormwater tunnel. A short introduction to modelling of stormwater flow, is therefore appropriate.

Consider an arbitrary region such as a city with well-defined geographical boundaries. In rainfall-runoff modelling, this region is referred to as a catchment. The goal is to model how a given distribution of rainfall over time runs off to one or more locations of interest within the catchment. The graph of water volume (or sometimes water level) over time at a location of interest is commonly called a hydrograph, although in this thesis, it will instead be denoted the rainfall-response, see Fig. 3.1. In the case of Paper A, there is just one location of interest. which is the downstream end of the Damhus stormwater tunnel.

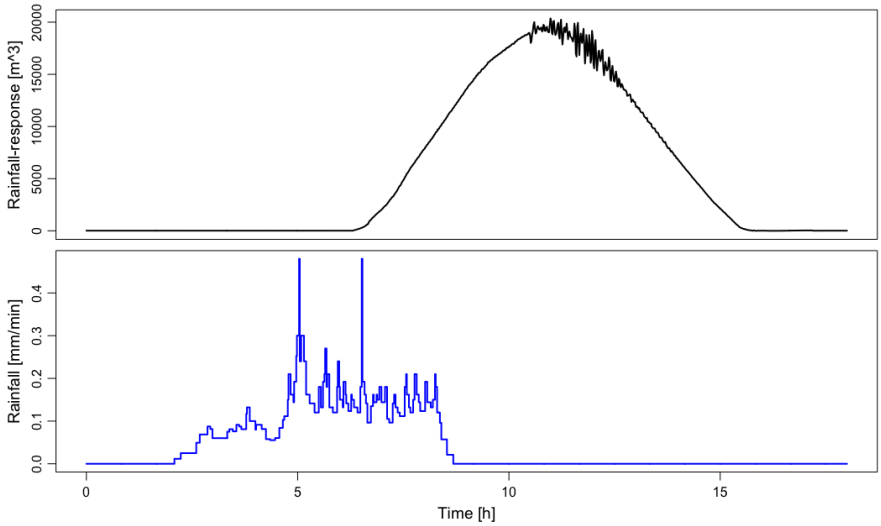

Figure 3.1: An example of rainfall-response (or hydrograph). In this case, the retention time is several hours.

3.1 The linear reservoir model

A classical way of modelling rainfall-response is to divide the catchment-sewer-system into a discrete number of sections, and then model the mass transfer between adjacent sections using differential equations. For example, if X_1 and X_2 represent two adjacent sections, then the mass transfer from X_1 to X_2 can be described by

$$\begin{aligned} \frac{\mathrm{d}X_{1,t}}{\mathrm{d}t} &= -rX_{1,t} \\ \frac{\mathrm{d}X_{2,t}}{\mathrm{d}t} &= rX_{1,t}, \end{aligned} \tag{3.1}$$

where r is rate of transfer. This description is essentially a series of conceptual reservoirs, and since the equations are linear differential equations, the model is called a linear reservoir model (Pedersen et al., 1980). The order of the model is denoted n, which is equal to the number of conceptual reservoirs in the model (not counting the final section which represents the location of interest). While higher n generally yields a better approximation to the true rainfall-response, a practical model can be achieved with relatively few reservoirs, usually around 2-12, see Fig. 3.2.

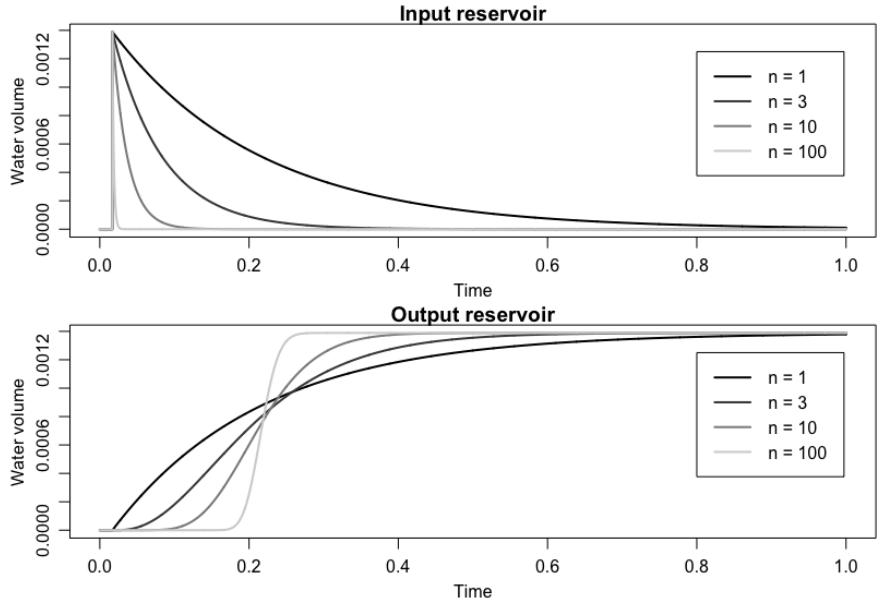

Figure 3.2: Comparison of the water volumes at the input (upstream) and output (downstream) in linear reservoir models with different n. All models have time constant $K = 0.2$.

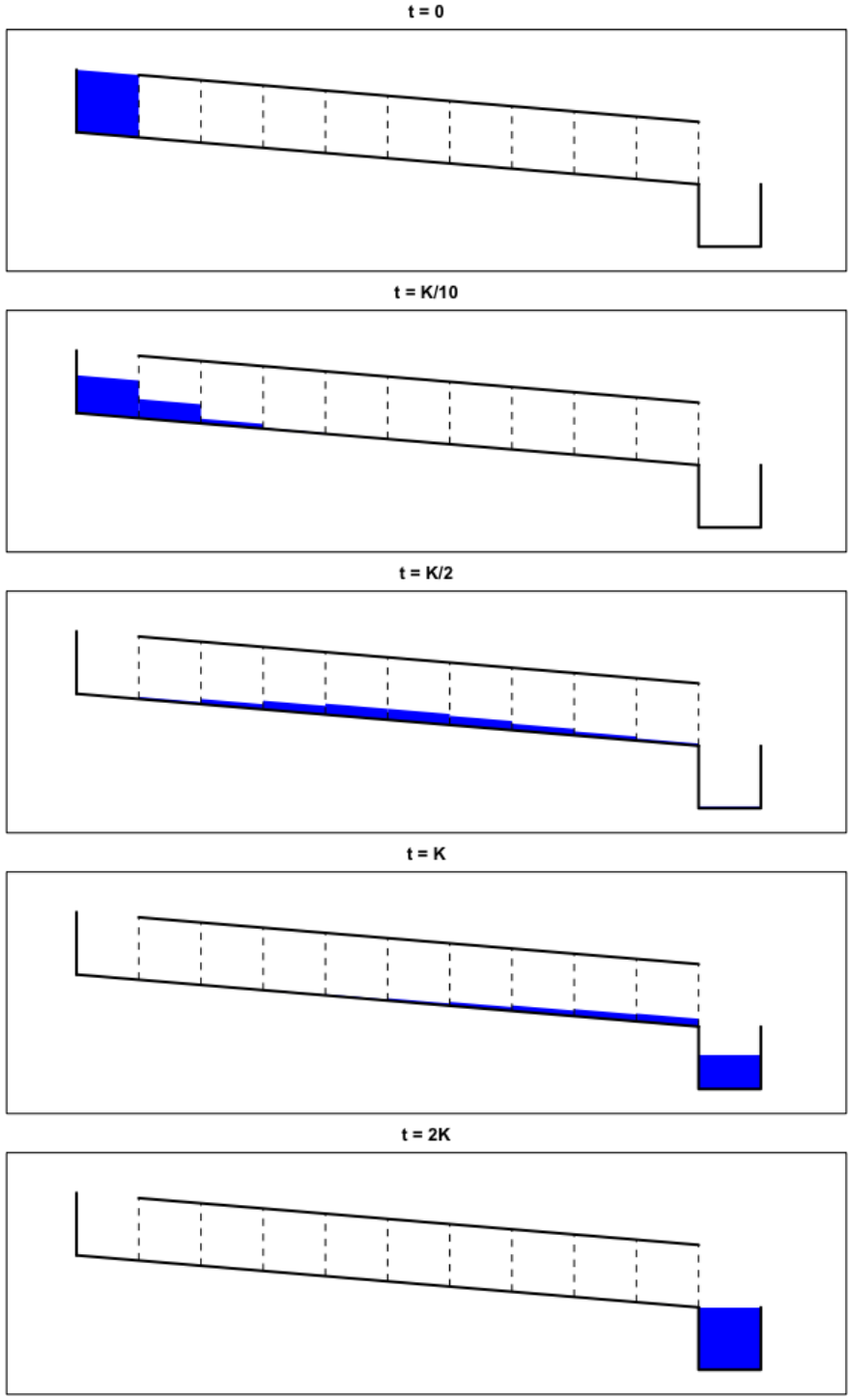

Figure 3.3: Linear reservoir model with $n = 10$ reservoirs. All the water is contained in the leftmost (upstream) reservoir at time $t = 0$ and then immediately starts transferring through the series of reservoirs towards the rightmost (downstream) reservoir.

n	Average retention time
1	0.19276
2	0.19949
3	0.19996
4	0.20000
10	0.20000
100	0.20000

Table 3.1: Calculated average retention times for linear reservoir models with $K = 0.2$ and various values of n. Indeed, the average retention approaches K very fast for increasing n.

Often, the substitution $r = n/K$ is used. Then, K is a *time constant* equal to the average retention time, i.e. the average time it takes for the rainfall to travel from the point where it hits the catchment surface until it reaches the point of interest, see Table 3.1. Clearly, a large K translates to a slow rate of transfer, and hence a long retention time, while conversely a small K translates to a fast runoff with short retention time. Fig. 3.3 shows an example of a linear reservoir model of a pipe consisting of $n = 10$ reservoirs with the distribution of rainfall-runoff after different amounts of time elapsed. For example, after $t = K/2$, the water is distributed around the half-way point between input and output, and after $t = K$, roughly half of the water has reached its final destination.

3.2 Integration of linear reservoir models into the CTSM framework

While the CTSM (see Chapter 2) serves as the skeleton for grey-box modelling, the linear reservoir concept provides the physics needed to make the CTSM resemble specifically rainfall-response. The integration of the two frameworks has been demonstrated by e.g. Breinholt et al. (2011) and is pretty straightforward. Consider, for example, a linear reservoir model with time constant K and $n = 2$. The rainfall input can be modelled as precipitation U_t measured in m/h multiplied by the catchment area A measured in m^2, and be added to the first equation. Hence, the model becomes:

$$\begin{aligned} \frac{\mathrm{d}X_{1,t}}{\mathrm{d}t} &= AU_t - \frac{2}{K}X_{1,t} \\ \frac{\mathrm{d}X_{2,t}}{\mathrm{d}t} &= \frac{2}{K}X_{1,t} - \frac{2}{K}X_{2,t} \\ \frac{\mathrm{d}X_{3,t}}{\mathrm{d}t} &= \frac{2}{K}X_{2,t}. \end{aligned} \tag{3.2}$$

Indeed, the rainfall input is a volume per time, as should be expected in a mass transfer equation, in this case measured in m^3/h. In practice, the choice of units depends on convenience.

All there is left to do now is to "upgrade" Eq. (3.2) to a CTSM by moving $\mathrm{d}t$ to the right-hand side, add some diffusion terms and include an observation equation. For instance, by borrowing the structure from Eq. (2.4), the resulting CTSM could look like the following:

$$\begin{aligned}
\mathrm{d}X_{1,t} &= (AU_t - \frac{2}{K}X_{1,t})\mathrm{d}t + \sigma \mathrm{d}W_{1,t} \\
\mathrm{d}X_{2,t} &= (\frac{2}{K}X_{1,t} - \frac{2}{K}X_{2,t})\mathrm{d}t + \sigma \mathrm{d}W_{2,t} \\
\mathrm{d}X_{3,t} &= \frac{2}{K}X_{2,t}\mathrm{d}t + \sigma \mathrm{d}W_{3,t} \\
Y_k &= X_{3,t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2).
\end{aligned} \tag{3.3}$$

This concludes the methodological chapters and in turn the basic foundation necessary for Chapter 4 - the modelling of the rainfall-response in the Damhus stormwater tunnel.

Chapter 4

Development of the SDE-based rainfall-runoff model

4.1 Overview

This chapter intends to document the development of the rainfall-response forecasting model which was published in Paper A. As prepared by the two preceding chapters, this is a CTSM that inherits its physical structure from the linear reservoir model concept. It models the rainfall-response in a Danish stormwater tunnel, which is located within the Damhus catchment in Copenhagen.

Paper A already documents the reasoning behind the final model, how it was estimated and what its forecasting capabilities are. However, the actual modelling process from the first attempt to the published version has been a long, iterative process with a series of obstacles that had to be dealt with. I believe that this model development process and its challenges are not unique to this specific case study, but will reappear in some form in many future modelling scenarios for all eternity. Therefore, this chapter focuses on the model development process and its challenges, and keeps the detailed reporting on the final model within Paper A itself.

4.1.1 The Damhus urban drainage case

The Damhus catchment is a 47 km^2 large urban area located in Copenhagen. It contains a combined sewer system, in which both household sewage and rainfall-runoff is accumulated from all over the catchment and passed along to a wastewater treatment plant (WWTP) downstream. In addition, a dedicated stormwater tunnel, 'the Damhus tunnel', is connected to the combined sewer system via a number of overflow structures. The tunnel adds extra storage capacity to the

overall sewer system, intended to contain large amounts of stormwater during heavy rain events and hence reduce the risk and frequency of floods in the urban area. Finally, a storage tower with a pumping facility, 'the Bottle Bridge', is installed at the downstream end of the tunnel. From here, the stormwater can be pumped back to the downstream part of the combined sewer system and ultimately led to the WWTP, just like the everyday sewage. The drainage system is sketched in Fig. 4.1.

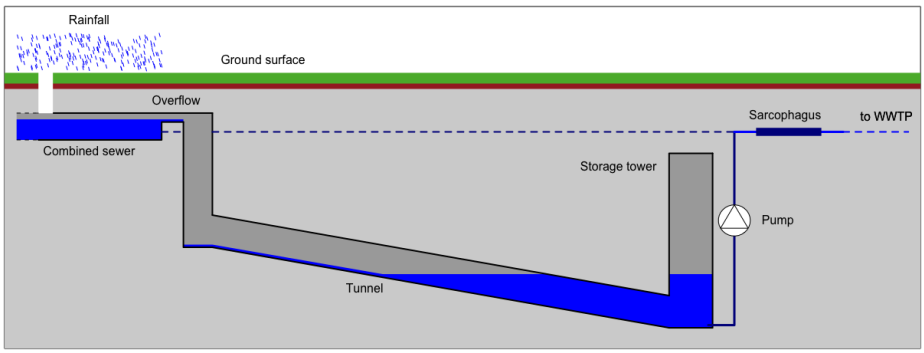

Figure 4.1: Conceptual drawing of the Damhus system with its key elements.

4.1.2 Why the Damhus case is worth modelling

The Damhus system as a case study for grey-box modelling is interesting for two reasons.

1. The system carries potential for an application of MPC to the pumping operation, which can cut expenses. This is discussed in Paper A, and will not be repeated in this thesis.
2. The system has significant non-linearities in the relationship between input (rainfall) and response (stormwater in the tunnel). The existing literature on grey-box models applied to urban drainage has, to my knowledge, not covered how to deal with this.

Paper A provides a solution to the second problem. It is case-specific, but while the paper does not cover how the solution can be generalized to other drainage systems, the principle is reasonably simple and can likely be applied to other cases after some modification.

Loosely spoken, if the system was completely linear, then more rainfall would always equal more stormwater in the tunnel. This is not the case, as the system does have non-linearities, see Fig. 4.2. First, the tunnel has a maximum capacity, which means that if the tunnel is already filled, then additional rainfall will not translate into more stormwater in the tunnel. Secondly, no water will enter the tunnel until a certain water level, a crest level, in the combined sewer system is met. At that point, water will start overflowing from the combined sewer system into the tunnel. As a result, the rainfall-response will be close to zero for smaller rain events and then increase dramatically when a rain event is sufficiently intense or long-lasting. Both the maximum capacity and the overflow crest hence constitute natural non-linearities in the system. The research in Paper A focuses on dealing with the latter non-linearity.

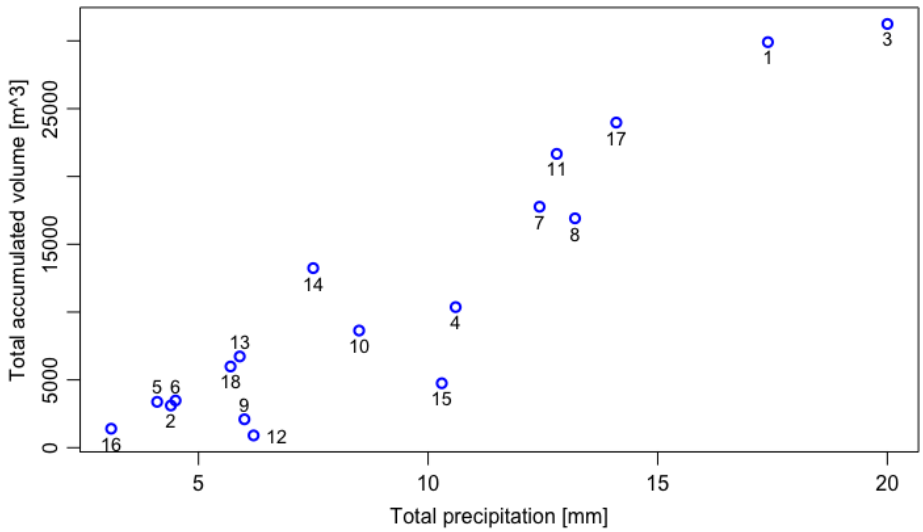

Figure 4.2: Comparison of 18 rainfall events in terms of total precipitation vs. total amount of stormwater ending up in the Damhus tunnel. While events with higher total precipitation does generally equate more water in the tunnel, the relationship is not linear. Event ID numbers are attached to the points.

4.1.3 Data

A thorough description of the data used for the Damhus case is embedded in Paper A, including characteristics of the raw as well as post-processed data. Below follows a minimal overview of the post-processed data which are used for the modelling of rainfall-response. Three different variables are needed, namely:

1. Volume observations in m^3, y_t, (post-processed water level observations, applied in units of 1000 m^3).
2. Rainfall observations in mm/min, U_t, (average rainfall intensity computed from two rain gauges in the catchment, applied in units of mm/h).
3. Pumping data in m^3/min, P_t (applied in units of 1000 m^3/h).

All of the above are available in 1-minute resolution, although a conversion to 5-minute resolution is also used later in the modelling process. Altogether, the dataset consists of 18 rainfall events, featuring a wide selection of different intensities and duration, which is convenient for estimating a model intended for reliable forecasting of any rainfall event. For example, event no. 7 is made up by two rainfall events in quick succession, as is seen in Fig. 4.3. All 18 rainfall events are visualized in Appendix A, see Fig. A.1, A.2 and A.3.

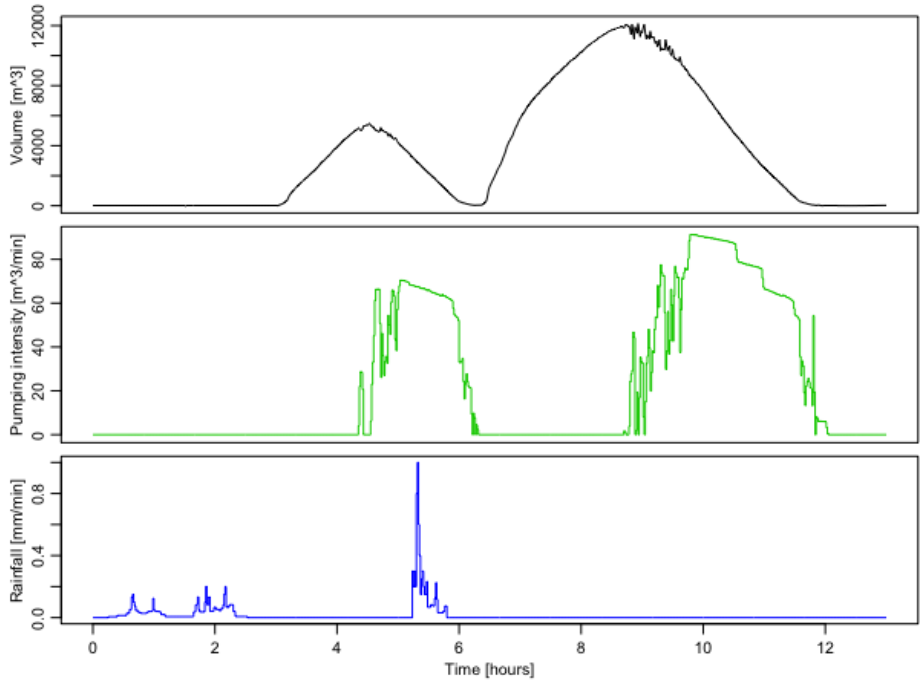

Figure 4.3: Post-processed data of event no. 7. Notice how the pumping activity coincides with water seemingly disappearing from the tunnel.

4.1.4 Outline of the modelling progression

Section 4.2 documents the detailed development of the CTSM, which is executed in 12 steps altogether. Every step is reported in terms of its system equations (see Section 2.2), parameter estimates and a brief graphical representation of model performance. In order to make it as easy as possible to follow the progression, the 12 steps may be boiled down to the following three blocks:

- Step 1-2: To start out, a naive CTSM with purely linear equations and state-independent diffusion is formulated (Fig. 4.4 top).
- Step 3-8: It is recognized that a purely linear model will not suffice to explain the long response delay. Consequently, a sigmoid function is introduced as a representation of the overflow crest (Fig. 4.4 middle).
- Step 9-12: When the physical structure of the model is satisfactory, the diffusion is upgraded to a state-dependent one in order to achieve a realistic distribution of forecast uncertainty (Fig. 4.4 bottom).

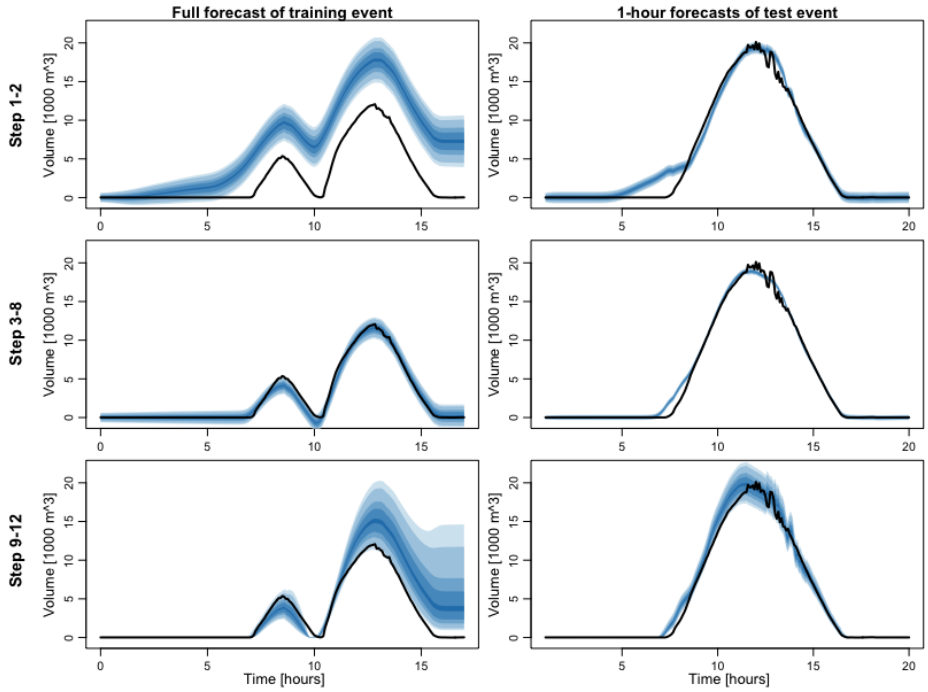

Figure 4.4: Probabilistic forecasts of rainfall-response issued by the models in step 2 (top row), 8 (middle row) and 12 (botttom row), respectively. The left and right columns feature forecasts on training data and test data, respectively.

4.2 Model development - step by step

4.2.1 Step 1 - the first linear reservoir model

The linear reservoir model embedded in a CTSM has already been introduced and discussed in Chapter 3. In the context of the Damhus system, it is convenient to consider the form in Eq. (3.3) with a couple of modifications. First, in that example $n = 2$ was used for simplicity, but in the practical case, we choose $n = 4$, because a higher model order makes it easier to get a smooth fit. Secondly, recall that accumulated water is eventually pumped out of the system (see Fig. 4.3), and this needs to be captured by the model. The pumping signal is denoted P_t, measured in the same units as the system states, and is subtracted from the last system equation. Hence, the first model is a CTSM with 5 states and the following system equations,

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{4}{K}X_{1,t} \\ \frac{4}{K}X_{1,t} - \frac{4}{K}X_{2,t} \\ \frac{4}{K}X_{2,t} - \frac{4}{K}X_{3,t} \\ \frac{4}{K}X_{3,t} - \frac{4}{K}X_{4,t} \\ \frac{4}{K}X_{4,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_3 \mathrm{d}W_{3,t} \\ \sigma_4 \mathrm{d}W_{4,t} \\ \sigma_5 \mathrm{d}W_{5,t} \end{pmatrix} \tag{4.1}$$

and the observation equation,

$$Y_k = X_{5,t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2). \tag{4.2}$$

This system is characterized by one time constant, K, and the rate of transfer from one state to another is thus $4/K$. Ideally, the pumping signal, P_t, would be modelled as a function of the system states in accordance with the actual existing real-time control scheme. However, this has proved impossible to reconstruct from the data. Instead, the measured P_t is simply used, independently of the system states. Such a simplification is expected to be sufficient for parameter estimation.

Until a good model structure has been identified, only one dataset will be used for parameter estimation, specifically event no. 7 (see Fig. 4.3). Using the estimation framework outlined in Chapter 2, the model parameters are estimated to

A	K	σ_1	σ_2	σ_3	σ_4	σ_5	σ_ε
1.965	1.944	0.01	0.01	0.01	1.285	0.01	0.1

The behavior of the model can be examined in Fig. 4.5. The figure displays 10 random simulations (blue) of the fitted model given only initial conditions

($X_i = 0$ for $i = 1..5$) and external variables (rainfall and pumping signal). The true volume observations are shown in black.

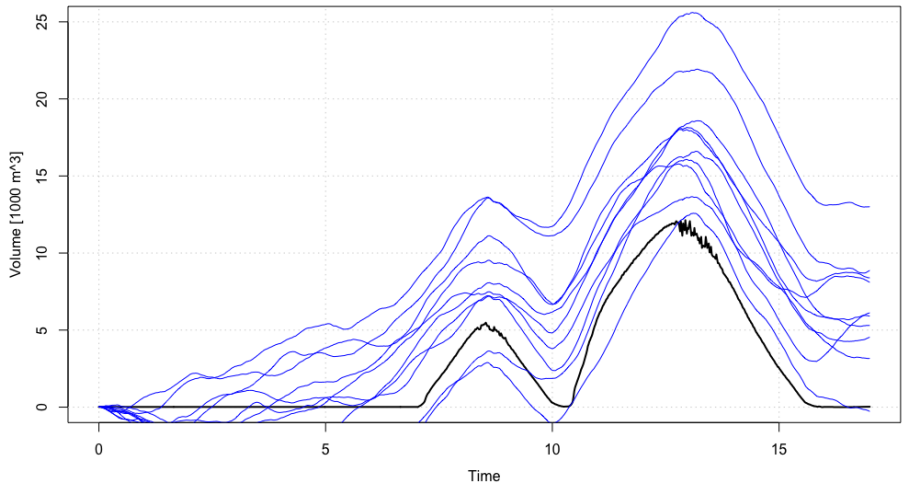

Figure 4.5: 10 realizations of event no. 7 by the CTSM estimated in step 1 (blue), compared with the observed volume (black).

While the camel shape of the volume propagation is mimicked to some extent, the variance seems extremely large. This is because the overall system variance of the model has been estimated as very high, with pretty much everything thrown into σ_4. Also, noting that several ensemble members dive into negative volumes, it is clear that the physics are not well-captured by this model. The estimation is essentially compensating for a weak physical model structure by attributing most of the variation in the data to system noise.

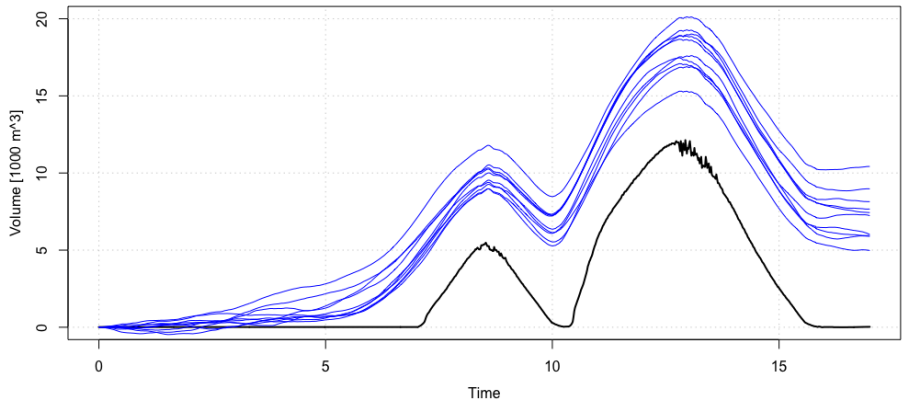

Figure 4.6: 10 realizations of event no. 7 by the CTSM estimated in step 1 with $\sigma_4 = 0.4$ (blue), compared with the observed volume (black).

This can easily be confirmed by lowering σ_4 to constrain the system noise. In Fig. 4.6, the same model with the same parameter estimates, except σ_4 set to 0.4, is shown. This leaves no doubt that the physical description is completely off, because way too much water is flowing into the system too early.

Before moving on to improve this seemingly deficient model structure, we will limit each of the system noise parameters to at most 0.1, to see if that could force the estimation to put more emphasis on the physics. The result is shown in Fig. 4.7.

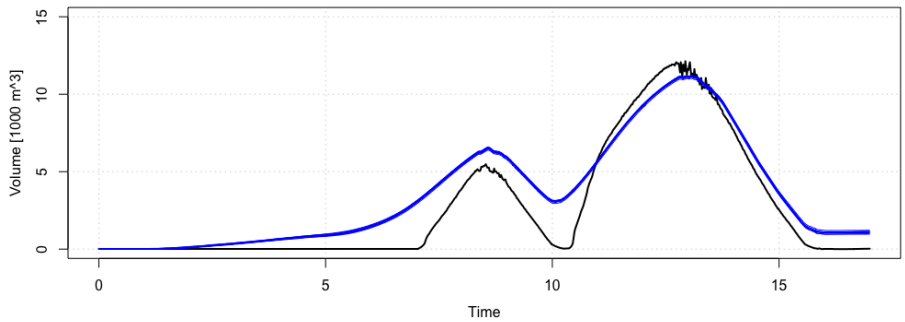

Figure 4.7: 10 realizations of event no. 7 by the CTSM in step 1 refitted with low maximal variance, compared with the observed volume (black). Note that because of the low variance, the 10 realizations shown in blue are almost identical to each other and appear as one thick curve.

Not surprisingly, there is almost no variance in the model anymore, which has indeed forced the prediction averages to come physically closer to the observations. However, it is seen that this happens at the expense of the camel shape. The peaks are closer to reality, but the valleys are way off, making the slopes flatter than desired. We conclude that the model has some promising features but needs to be extended.

4.2.2 Step 2 - introducing additional time constants

The model in step 1 was able to mimic the camel shape, but failed to capture the time delay from rainfall to response. For the second step, it is hypothesized that having more than one time constant may improve the latter issue. With only one time constant, it is assumed that the water travels with the same rate everywhere in the system. This should not be the case, rather the water would travel with different rates over the ground surface, through the combined sewer system and in the tunnel, respectively. Therefore, we attach a unique time constant to each of the four mass transfers (changes w.r.t Eq. (4.1) are

highlighted in teal) and get:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{1}{K_1}X_{1,t} \\ \frac{1}{K_1}X_{1,t} - \frac{1}{K_2}X_{2,t} \\ \frac{1}{K_2}X_{2,t} - \frac{1}{K_3}X_{3,t} \\ \frac{1}{K_3}X_{3,t} - \frac{1}{K_4}X_{4,t} \\ \frac{1}{K_4}X_{4,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_3 \mathrm{d}W_{3,t} \\ \sigma_4 \mathrm{d}W_{4,t} \\ \sigma_5 \mathrm{d}W_{5,t} \end{pmatrix}. \tag{4.3}$$

The model parameters are estimated to

A	K_1	K_2	K_3	K_4	σ_1	σ_2	σ_3	σ_4	σ_5	σ_ε
1.974	0.574	0.574	0.396	0.403	0.01	0.01	1	0.959	0.1	0.1

It is seen that $K_1 \approx K_2$ and $K_3 \approx K_4$. Thus, essentially 2 distinct time constants are identified under this model structure. It is concluded that one time constant is generally not enough. The exact amount may be subject to change in future iterations. The sum of the time constants is 1.944, i.e. effectively the same as $K = 1.947$ in step 1. The overall model performance is not improved with respect to step 1, as evident by Fig. 4.8.

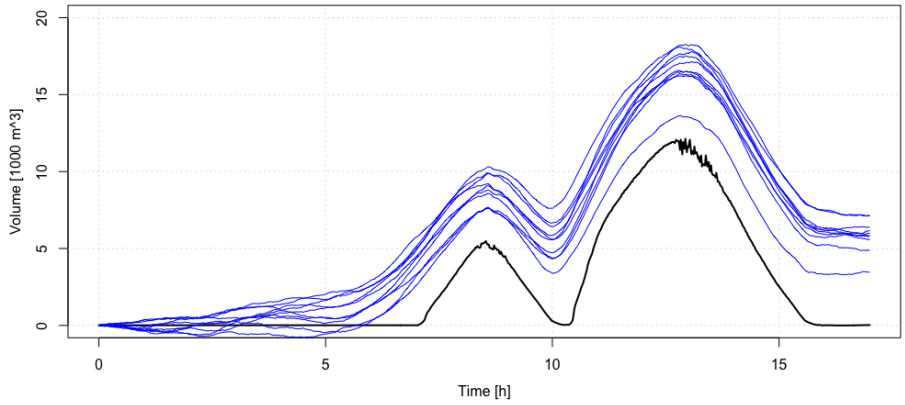

Figure 4.8: 10 realizations of event no. 7 by the CTSM estimated in step 2 with (blue), compared with the observed volume (black). The realizations are produced with $\sigma_3 = 0.01$ and $\sigma_4 = 0.4$ for the same reasons as for Fig. 4.6

4.2.3 Step 3 - modelling the overflow crest as a sigmoid function

It is clear that increasing the model order and using multiple time constants does not suffice to capture the time delay from rainfall to response. To find

a solution, we consider the system layout again, see Fig. 4.1. While there is a steady flow from the ground surface to the combined sewer system, nothing really happens in the tunnel before a certain water level threshold is met, at which point the water finally starts flowing into the tunnel. This threshold is called an overflow crest.

If the overflow crest can be properly integrated in the model structure, the rainfall-response time delay should be captured better than in the previous steps. To accomplish this, we introduce the abundantly used sigmoid function (Berkson, 1953),

$$q(x) = \frac{1}{1 + \mathrm{e}^{-\alpha(x-\beta)}} \tag{4.4}$$

where α and β are the sharpness and threshold of the function, respectively. Consider then two arbitrary sewer states separated by an overflow crest, X_{pre} and X_{post}, with the following mass transfer relationship,

$$\begin{aligned} \mathrm{d}X_{\mathrm{pre},t} &= (\omega_t - \frac{1}{K} X_{\mathrm{pre},t})\mathrm{d}t \\ \mathrm{d}X_{\mathrm{post},t} &= q(X_{\mathrm{pre,t}}) \frac{1}{K} X_{\mathrm{pre},t} \mathrm{d}t, \end{aligned} \tag{4.5}$$

where ω_t is some arbitrary inflow to X_{pre}. The inflow to X_{post} from X_{pre} is thus limited by $q(x)$ roughly until $x \geq \beta$. This effect is illustrated in Fig. 4.9, where it is indeed seen that the water only starts flowing in around the time at which the threshold β is reached in X_{pre}.

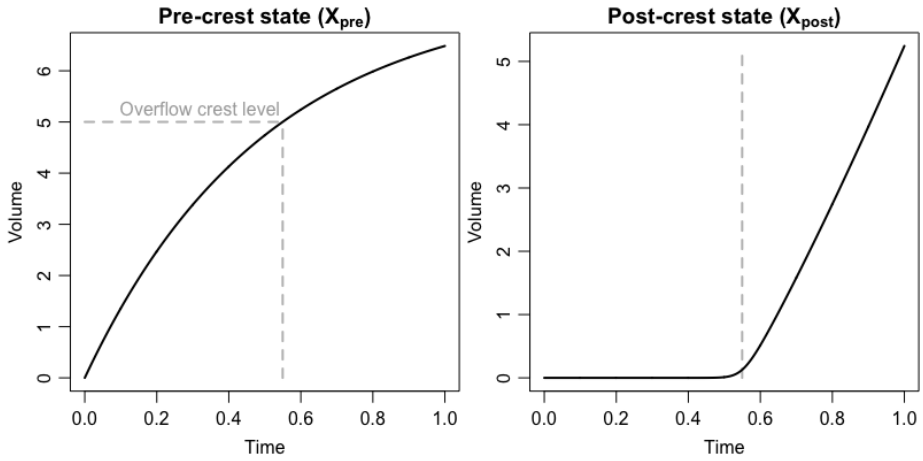

Figure 4.9: The system in Eq. 4.5 realized with $K = 0.5$, $\alpha = 10$ and $\beta = 5$. The time at which $X_{\mathrm{pre}} = \beta$ is marked by gray dashed lines.

A natural way to integrate this crest representation into the Damhus system equations, is to regard K_2 as the time constant for the tunnel, and K_1 as the time constant for the flow leading up to that point. It follows that the overflow crest is located between X_3 and X_4, and hence, the mass transfer between these two states is multiplied by $q(X_3)$. The resulting system equations are as follows,

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{2}{K_2}X_{3,t} \\ q(X_3,t)\frac{2}{K_2}X_{3,t} - \frac{2}{K_2}X_{4,t} \\ \frac{2}{K_2}X_{4,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_3 \mathrm{d}W_{3,t} \\ \sigma_4 \mathrm{d}W_{4,t} \\ \sigma_5 \mathrm{d}W_{5,t} \end{pmatrix}. \tag{4.6}$$

α and β are estimated in the optimization routine along with the other parameters. Note that for numerical reasons, it is important that the product of α and β is never too high, because it can cause the exponential part of $q(x)$ to explode and crash the optimization routine. The model parameters are estimated to

A	α	β	K_1	K_2	σ_1	σ_2	σ_3	σ_4	σ_5	σ_ε
2.672	24.286	0.821	2.119	0.453	0.1	0.1	0.1	0.1	0.1	0.1

It is seen that both A and the sum of the time constants have increased compared to the previous steps. Furthermore, $K_1 = 2.119$ is much larger than $K_2 = 0.453$, which means that it takes much longer for the water to reach the crest level than to flow through the tunnel afterwards. The realization of the model is shown in Fig. 4.10, and for the first time, the time delay is really well captured while the camel shape is kept intact simultaneously. Indeed, introducing the crest function to the system is working as intended.

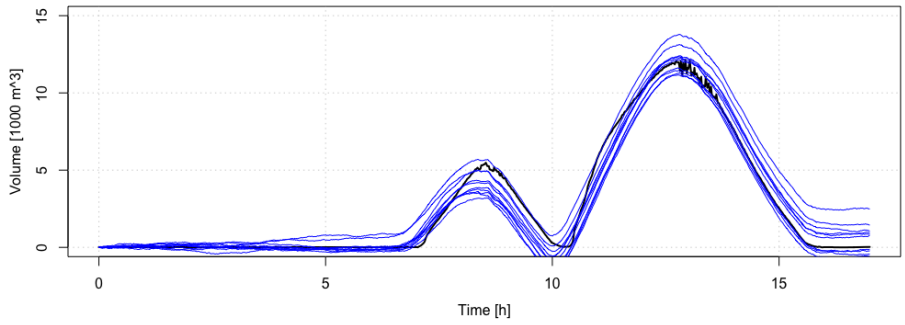

Figure 4.10: 10 realizations of event no. 7 by the CTSM estimated in step 3 (blue), compared with the observed volume (black).

4.2.4 Step 4 - reduction of parameter space

The model identified in step 3 is very promising. However, it has quite many parameters. In particular, having 5 diffusion parameters which all get the same estimate, might be overkill. Therefore, for step 4, we are going to test whether the number of diffusion parameters can be reduced. It is compelling to imagine that the first system equation should be very noisy due to being directly affected by rainfall, a notoriously uncertain and volatile type of input. On the other hand, the rest of the system is just a collection of mass transfers in a closed system with no further external inputs. Thus, the latter could be considered under one shared diffusion parameter, σ_2, while leaving σ_1 as unique. The corresponding system equations become:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{2}{K_2}X_{3,t} \\ q(X_3,t)\frac{2}{K_2}X_{3,t} - \frac{2}{K_2}X_{4,t} \\ \frac{2}{K_2}X_{4,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_2 \mathrm{d}W_{3,t} \\ \sigma_2 \mathrm{d}W_{4,t} \\ \sigma_2 \mathrm{d}W_{5,t} \end{pmatrix}. \tag{4.7}$$

The physical parameter estimates of this model compared to the ones from step 3 are virtually unchanged:

	A	α	β	K_1	K_2
Unique σ_i for each X_i	2.67237	24.28629	0.82131	2.11916	0.45284
Only σ_1 and σ_2	2.67237	24.28596	0.82131	2.11915	0.45285

Furthermore, both models have a negative log-likelihood of -30.74976 and can thus safely be regarded as equally performing. There is therefore no reason to retain the extra diffusion parameters, and the system in Eq. (4.7) is hence preferred over that in Eq. (4.6).

4.2.5 Step 5 - a revisit to time constants

After the crest function has been introduced to the system, the model has fundamentally changed from a linear to a non-linear one. Therefore, the previous finding that the system only has two different time constants should be reconfirmed. As in step 2, we now associate each mass transfer with its own unique

time constant, and the resulting system becomes:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{1}{K_1}X_{1,t} \\ \frac{1}{K_1}X_{1,t} - \frac{1}{K_2}X_{2,t} \\ \frac{1}{K_2}X_{2,t} - \frac{1}{K_0}X_{3,t} \\ q(X_3,t)\frac{1}{K_0}X_{3,t} - \frac{1}{K_3}X_{4,t} \\ \frac{1}{K_3}X_{4,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_2 \mathrm{d}W_{3,t} \\ \sigma_2 \mathrm{d}W_{4,t} \\ \sigma_2 \mathrm{d}W_{5,t} \end{pmatrix}. \tag{4.8}$$

Here, K_1, K_2 and K_3 are thought of as the time constants for the flow on the ground surface, the combined sewer system and the tunnel, respectively, while K_0 refers to the overflow from the combined sewer to the tunnel.

The model parameters are estimated to

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
2.676	10.089	1.615	1.07	0.879	0.136	0.431	0.1	0.1	0.1

It is seen that all 4 time constants attain unique values. Furthermore, the negative log-likelihood is -38.980 and hence slightly better than in the step 4. It is concluded that all four time constants should be kept in the model going forward.

4.2.6 Step 6 - selection of the number of states

With the time constant setup sorted, the next step is to determine the order of the model sections associated with each time constant. Consider first the simplest model where each section is of order 1 - this is equivalent to the model from step 5 (Eq. (4.8)). Then a classical forward selection process is applied (Blanchet et al., 2008), where a heuristic assessment of the difference in negative log-likelihood is used to determine whether two models are significantly different.

In the first iteration of the forward selection, three new models are estimated. In each model, one of the model sections associated with either of the three time constants, K_1, K_2 and K_3 respectively, has its model order increased from 1 to 2, while leaving the other model sections at order 1. The section associated with K_0 is always assumed to be of order 1. This yields the following negative log-likelihoods,

Order of (K_1, K_2, K_3)	ℓ
(1,1,1)	-38.980
(2,1,1)	-78.585
(1,2,1)	-77.362
(1,1,2)	-62.905

The best model is (2,1,1) with $\ell = -78.585$, which is deemed to be sufficiently better than (1,1,1) with $\ell = -38.980$, and therefore (2,1,1) is selected.

In the second iteration (2,1,1) is tested against (3,1,1), (2,2,1) and (2,1,2), however no heuristically acceptable improvement in the negative log-likelihood is found for any of those three new models, and hence we settle on the (2,1,1)-model, i.e. the system equations:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{1}{K_2}X_{3,t} \\ \frac{1}{K_2}X_{3,t} - \frac{1}{K_0}X_{4,t} \\ q(X_4)\frac{1}{K_0}X_{4,t} - \frac{1}{K_3}X_{5,t} \\ \frac{1}{K_3}X_{5,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_2 \mathrm{d}W_{3,t} \\ \sigma_2 \mathrm{d}W_{4,t} \\ \sigma_2 \mathrm{d}W_{5,t} \\ \sigma_2 \mathrm{d}W_{6,t} \end{pmatrix}. \tag{4.9}$$

All of the parameter estimates are quite different from step 5, although the relation $K_1 > K_2 > K_0 > K_3$ is preserved:

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
2.729	16.540	0.568	1.821	0.383	0.067	0.141	0.1	0.1	0.1

4.2.7 Step 7 - reduction of sampling rate

At this point we are getting close to a final physical model structure, but before moving on to the development of the stochastic part, we shall attempt to trim the optimization load a bit.

Until this point, the original 1-minute resolution of the data has been used. However, in step 7, we will reduce to a 5-minute resolution (see Fig. 4.11), which gives the advantage that every evaluation of the negative log-likelihood contains 5 times fewer one-step predictions, and in turn potentially improves the runtime up to 5-fold. It is found that using 5-minute resolution, the runtime of the estimation procedure is approximately 21.4 seconds, while it is 83.0 seconds when 1-minute resolution is used. The flat amount of time saved is of course a

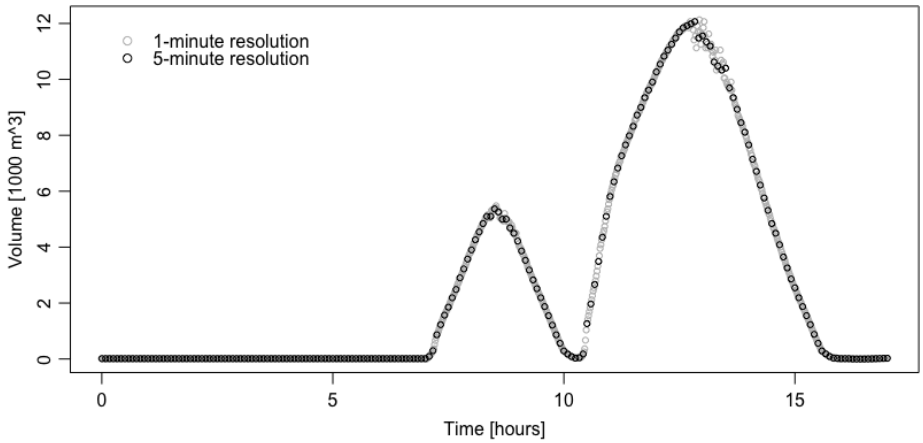

Figure 4.11: Comparison of 1-minute resolution (gray) vs. 5-minute resolution (black) of the rainfall-response in event no. 7. Here, it looks like the 5-minute resolution should be sufficient to capture the dynamics of the system.

CPU-specific result, however the magnitude of a ∼4-fold improvement should be representative regardless of the machine used for the computation.

The parameter estimates for the model with 5-minute resolution are as follows,

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
2.425	22.007	0.747	1.622	0.338	0.121	0.220	0.1	0.1	0.1

These are fairly similar to the corresponding estimates under the 1-minute resolution, and the model realization is still looking good, as seen in Fig. 4.12. Therefore, it is decided that a 5-minute resolution is good enough for model estimation and worth using over 1-minute resolution due to the time savings.

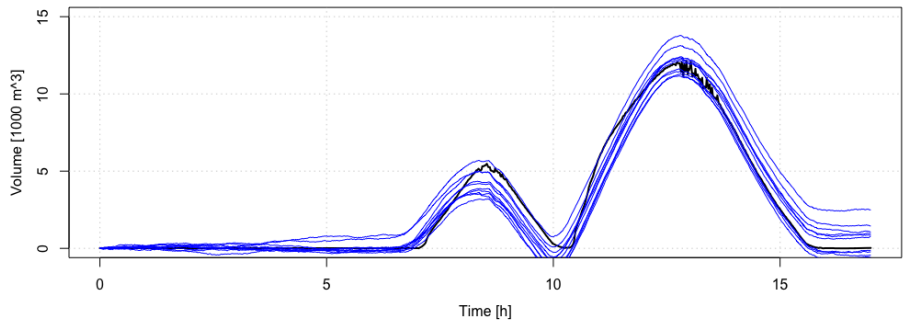

Figure 4.12: 10 realizations of event no. 7 by the CTSM estimated in step 7 (blue), compared with the observed volume (black). This is the first time a 5-minute resolution is used.

4.2.8 Step 8 - modifying the combined sewer wastewater flow

In this step, one final modification of the physical system structure is made. In a 100% logical model building order this would happen in an earlier step, however, chronologically speaking, this is how it happened and I am keen on keeping it as an 'oops-by-the-way'-step.

Consider the state equation for X_4 in Eq. (4.9). The outflow is here solely characterized by K_0 which is associated with the overflow from the combined sewer to the tunnel, and the equation essentially assumes that whatever does not go into the tunnel continues downstream through the combined sewer towards the wastewater treatment plant. However, the latter flow should not be assumed to have the same time constant as the overflow, but rather be associated with K_2 as the rest of the combined sewer. Therefore, the equation for X_4 is altered to include both of the two outflows, and the system becomes:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{2}{K_2}X_{3,t} \\ \frac{2}{K_2}X_{3,t} - (\frac{2}{K_2} + q(X_4)\frac{1}{K_0})X_{4,t} \\ q(X_4)\frac{1}{K_0}X_{4,t} - \frac{1}{K_3}X_{5,t} \\ \frac{1}{K_3}X_{5,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_2 \mathrm{d}W_{3,t} \\ \sigma_2 \mathrm{d}W_{4,t} \\ \sigma_2 \mathrm{d}W_{5,t} \\ \sigma_2 \mathrm{d}W_{6,t} \end{pmatrix}. \tag{4.10}$$

The most notable change in the parameter estimates is in A which is almost twice as large.

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
4.496	2.763	5.026	0.977	1.866	0.158	0.136	0.1	0.1	0.1

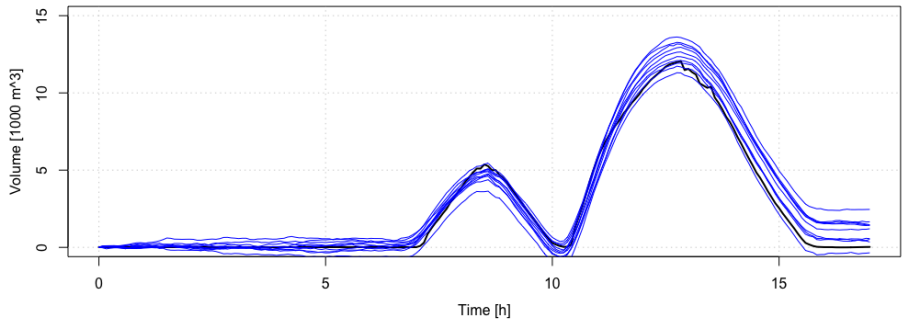

Figure 4.13: 10 realizations of event no. 7 by the CTSM estimated in step 8 (blue), compared with the observed volume (black).

The model realization is still looking reasonable, as seen in Fig. 4.13. All in all, the model in step 8 is not performing worse than previous models, while it has a more realistic representation of the combined sewer flow. Therefore, we proceed with this model for the next step.

4.2.9 Step 9 - introduction of state-dependent diffusion

A satisfactory physical structure has now been found. However, the uncertainty of the forecasts issued by the model is not very realistic. First, in Fig. 4.13 it is indicated that the spread is uniform regardless of whether there is no water or a lot of water in the tunnel. This is to be expected given the additive noise structure in Eq. (4.10). On the other hand, in a realistic model, there should not be much variance when there is little to no water in the tunnel, and considerably more variance when there is a lot of water. Secondly, it is clearly seen in Fig. 4.13 that the naive noise structure allows for the water volume forecast to sometimes attain negative values, which of course is absurd.

Both problems can be solved by switching to a system-dependent noise structure, where the diffusion scales with some function of water volume. There are several options for such a function, but we choose a direct scaling with the water volume in each respective state, i.e. the diffusion is $\sigma_1 X_{1,t} \mathrm{d}W_{1,t}$ for the first state and $\sigma_2 X_{i,t} \mathrm{d}W_{i,t}$ for $i = 2..6$. The updated system equations thus read:

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1} X_{1,t} \\ \frac{2}{K_1} X_{1,t} - \frac{2}{K_1} X_{2,t} \\ \frac{2}{K_1} X_{2,t} - \frac{2}{K_2} X_{3,t} \\ \frac{2}{K_2} X_{3,t} - (\frac{2}{K_2} + q(X_4)\frac{1}{K_0}) X_{4,t} \\ q(X_4)\frac{1}{K_0} X_{4,t} - \frac{1}{K_3} X_{5,t} \\ \frac{1}{K_3} X_{5,t} - P_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 X_{1,t} \mathrm{d}W_{1,t} \\ \sigma_2 X_{2,t} \mathrm{d}W_{2,t} \\ \sigma_2 X_{3,t} \mathrm{d}W_{3,t} \\ \sigma_2 X_{4,t} \mathrm{d}W_{4,t} \\ \sigma_2 X_{5,t} \mathrm{d}W_{5,t} \\ \sigma_2 X_{6,t} \mathrm{d}W_{6,t} \end{pmatrix}. \quad (4.11)$$

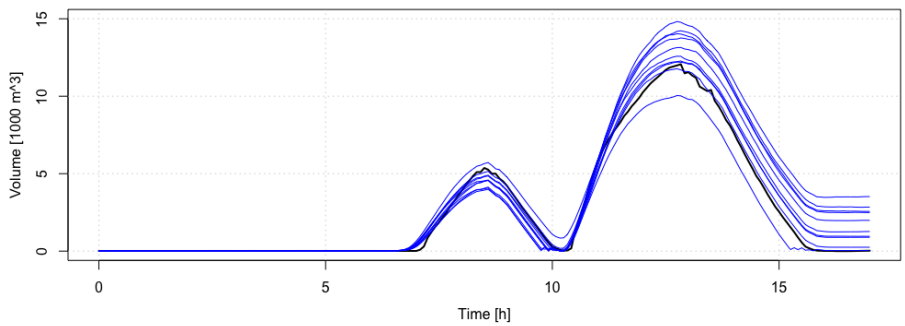

Figure 4.14: 10 realizations of event no. 7 by the CTSM stated in step 9 (blue), compared with the observed volume (black).

A quick test of the new model's potential is done by setting $\sigma_1 = 0.15$ and $\sigma_2 = 0.01$, while keeping the other parameter estimates from step 8 intact. A realization of the model is displayed in Fig. 4.14. Clearly, it is working as intended, as there is no longer any negative water, and the spread is extremely narrow for low water volumes while much wider for higher water volumes. A proper estimation of the parameters in the new model will take place in step 10-12.

4.2.10 Step 10 - applying the Lamperti transform

While the CTSM formulated in step 9 seems perfect for the case at hand, there is a minor holdup. The estimation algorithm in `ctsmr` relies on the extended Kalman filter which in turn requires the system noise to be Gaussian and hence state-independent (Breinholt et al., 2011). Therefore, this model cannot be estimated directly in its current form. Instead, we will consider a new set of random variables $Z_i = \log(X_i)$ for $i = 1..6$ and apply a Lamperti transform (Møller and Madsen, 2010) to Eq. (4.11) (see Paper A, supplementary material for the detailed calculation). The resulting system is as follows,

$$\mathrm{d}\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \end{pmatrix}_t = \begin{pmatrix} AU_t\mathrm{e}^{-Z_{1,t}} - \frac{2}{K_1} - \frac{\sigma_1^2}{2} \\ \frac{2}{K_1}\mathrm{e}^{(Z_{1,t}-Z_{2,t})} - \frac{2}{K_1} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_1}\mathrm{e}^{(Z_{2,t}-Z_{3,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_2}\mathrm{e}^{(Z_{3,t}-Z_{4,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} - \frac{1}{K_0}q(\mathrm{e}^{Z_{4,t}}) \\ \frac{1}{K_0}q(\mathrm{e}^{Z_{4,t}})\mathrm{e}^{(Z_{4,t}-Z_{5,t})} - \frac{1}{K_3} - \frac{\sigma_2^2}{2} \\ \frac{1}{K_3}\mathrm{e}^{(Z_{5,t}-Z_{6,t})} - P_t\mathrm{e}^{-Z_{6,t}} - \frac{\sigma_2^2}{2} \end{pmatrix}\mathrm{d}t + \begin{pmatrix} \sigma_1\mathrm{d}W_{1,t} \\ \sigma_2\mathrm{d}W_{2,t} \\ \sigma_2\mathrm{d}W_{3,t} \\ \sigma_2\mathrm{d}W_{4,t} \\ \sigma_2\mathrm{d}W_{5,t} \\ \sigma_2\mathrm{d}W_{6,t} \end{pmatrix}. \tag{4.12}$$

This system form has state-independent noise and can thus be estimated with `ctsmr`. The original states can trivially be reconstructed by

$$X_i = \mathrm{e}^{Z_i}, \quad \forall i. \tag{4.13}$$

However, when we try to run the estimation algorithm on the system in Eq. (4.12), the process crashes almost immediately and `ctsmr` reports that it is unable to find a numerical ODE solution, as seen in the console snippet below:

```
  2:     137.21893:  9.90882 0.997006  1.23188  1.52008  1.26853 0.780076  1.11059 0.100000 0.100000 0.100000
       A     alpha      beta        K0        K1        K2        K3    sigma1    sigma2  sigmaObs
9.8714924 1.1123378 1.5663262 1.5576942 1.1732057 0.6181977 0.8926921 0.1000000 0.1000000 0.1000000

Error in predict.ctsmr(M, newdata = Data.i) :
  Unable to perform numerical ODE solution. Code: 90
```

Figure 4.15: Console snippet from the failed optimization of the CTSM as defined in step 10.

For the first time, the system is apparently infeasible. This problem is handled in the next step.

4.2.11 Step 11 - understanding and respecting physical domain restrictions

In step 10, the optimization routine crashed before converging, and we need to figure out why. The routine crashed already in its second iteration, when the set of model parameters shown in Fig. 4.15 were selected. Specifically, it means that given this set of parameters, the series of one-step predictions needed for the negative log-likelihood could not be computed. Hence, a good starting point for troubleshooting is to compute a partial series of one-step predictions up to the point where it crashes, and then perform a visual inspection of the computed partial series.

It turns out, that the series can be predicted up to and including the 188th one-step prediction, corresponding to 15.67 hours into the series. The series is shown in both the Lamperti transformed Z_6-domain and the original X_6-domain in Fig. 4.16.

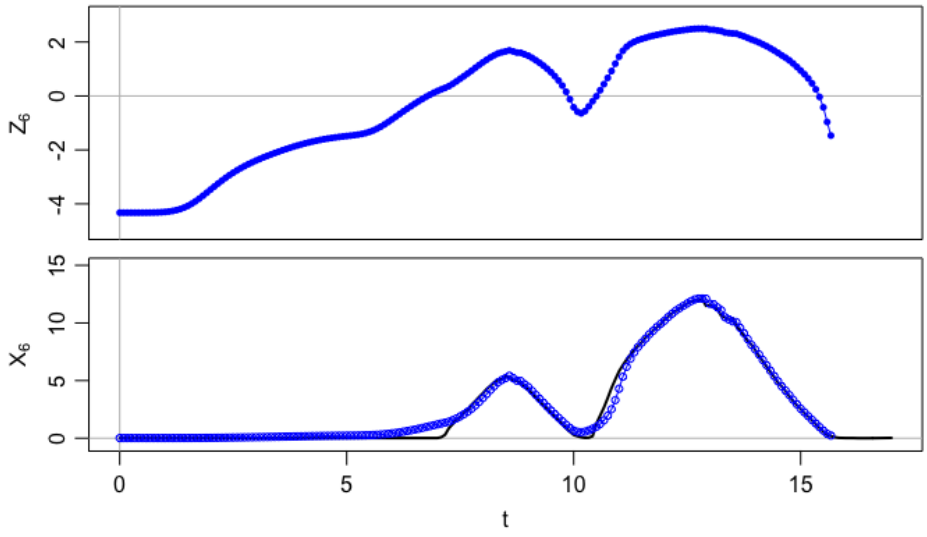

Figure 4.16: A series of one-step predictions (blue) produced by the CTSM from step 10 with the parameters from Fig. 4.15, compared with the real observations (black). The predictions are shown in both the Z_6-domain (top) and the X_6-domain (bottom).

Immediately, it is apparent that the last one-step predictions in the X_6-domain are very close to 0, and seem to be heading for the negative domain due to the relentless pumping activity taking place in that time interval. To make it easier to see, Fig. 4.17 zooms in on this time interval, where it indeed appears that the prediction series could break through the $X_6 = 0$ boundary within one or two time steps.

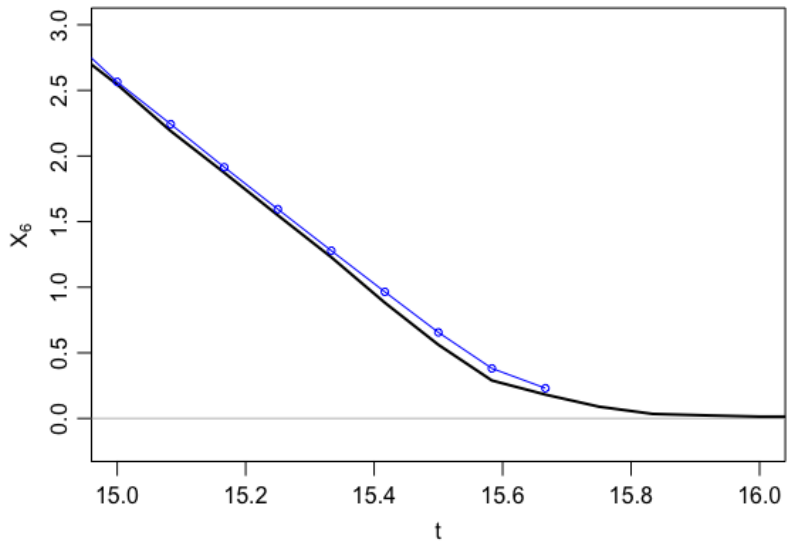

Figure 4.17: Magnified view of the critical time interval, within which the prediction routine crashes. The one-step predictions are shown in blue, and the observed water volume is shown in black.

Keep in mind that the Lamperti system was derived with in order to *avoid* exactly this physically absurd behaviour. Maybe, having a 'blind' pumping signal which does not refrain from pumping even when there is no water present in X_6, is breaking `ctsmr` in some way?

Having identified a potential flaw, we then consider a simplified ODE version of the system equation governing Z_6 (see Eq. (4.12)), where anything but the pumping-affected term is disregarded. Denoting the simplified state variable z, we have:

$$\frac{\mathrm{d}z}{\mathrm{d}t} = p \cdot \mathrm{e}^{-z}. \tag{4.14}$$

Solving this ODE for z with the initial condition $z(0) = z_0$ yields:

$$z = \log(p \cdot t + \mathrm{e}^{z_0}) \tag{4.15}$$

Hence, $p \cdot t + e^{z_0}$ must always be strictly positive. Since we want to consider a situation where water is being pumped out, we can for example put $p = -1$ and thus get

$$z = \log(-t + e^{z_0}). \tag{4.16}$$

A situation with little to no water present translates to e^{z_0} being small, which means z_0 is negative. Obviously, $-t+e^{z_0}$ will only remain positive before $t = e^{z_0}$, at which point there is a singularity. This is illustrated in Fig. 4.18 with $z_0 = 0$, and hence with the singularity in $t = 1$.

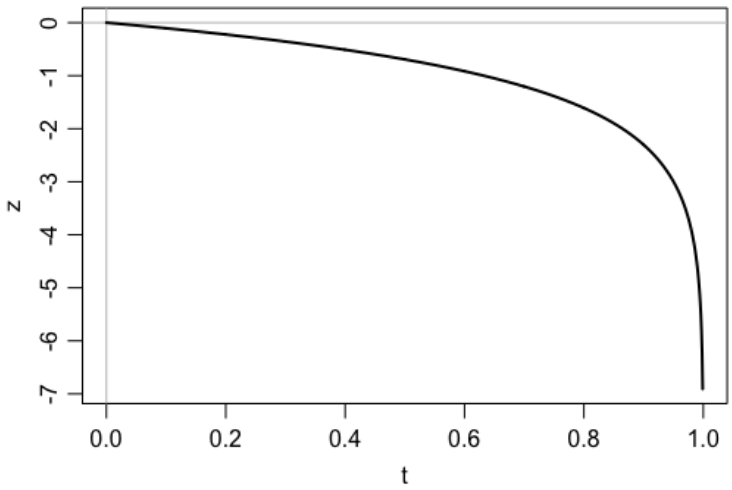

Figure 4.18: The evolution of $z = \log(-t + e^{z_0})$ with $z_0 = 0$.

Clearly, as the real water volume, e^z approaches 0, z approaches $-\infty$, and the time interval until the singularity is reached shortens dramatically. Then, if a one-step prediction is due to be computed, and the current pumping signal would cause the water volume in the original system to go below zero, it is now clear that even a very small time increment would cause $-t + e^{z_0}$ to be negative and thus cause Eq. (4.16) to break down.

It can be deduced that an analogous behaviour would happen in the Lamperti system in question, and it is hence concluded that pumping more water out of the system than is already present will inevitably cause the prediction routine in `ctsmr` to crash.

Therefore, it is necessary to safeguard the system equations such that the pumping signal will never cause the water volume to go below 0, no matter what. A

great solution to this problem is once again to use a sigmoid function, as we did in step 3 for the crest modelling. The new sigmoid function, $q_P(x)$, is given by

$$q_P(x) = \frac{1}{1 + \mathrm{e}^{-\alpha_P(x-\beta_P)}}, \tag{4.17}$$

with $\alpha_P = 200$ and $\beta_P = 0.05$ chosen. $q_P(x)$ is then multiplied with the pumping signal to attenuate the latter when the water volume becomes small. This effect is illustrated in Fig. 4.19.

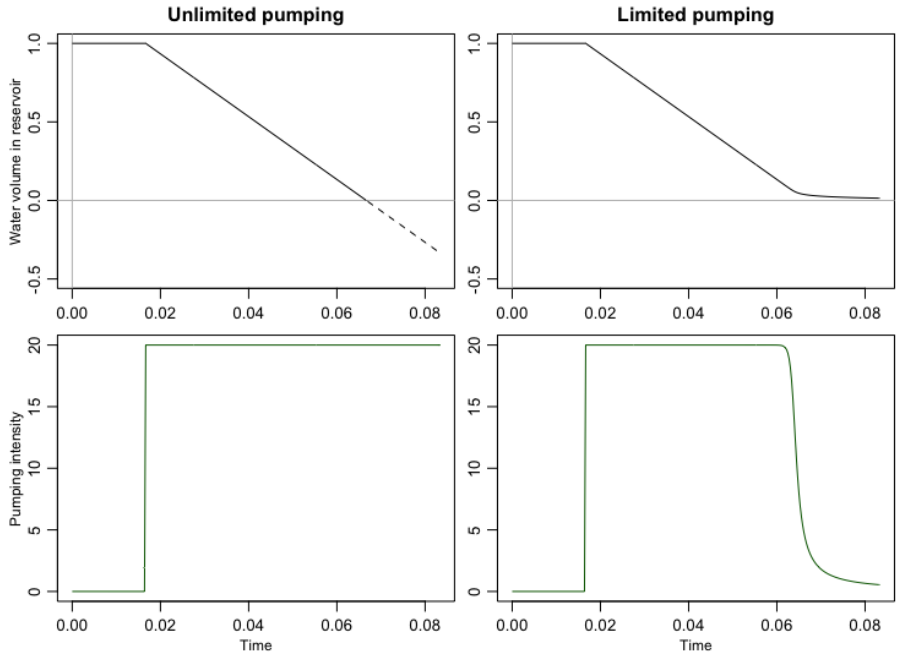

Figure 4.19: Theoretical water volume (top row) as a result of a constant pumping signal (bottom row) with a comparison of an unrestricted system (left column) vs. a restricted system (right column). In the restricted system, the pumping signal is multiplied by $q_P(x)$ and is thus increasingly attenuated for low values of water volume, which in turn ensures that no more water than what is present can be pumped out of the system.

Applying this solution to the system from step 10 changes the system equation for Z_6 to:

$$\mathrm{d}Z_{6,t} = (\frac{1}{K_3}e^{(Z_{5,t}-Z_{6,t})} - q_P(\mathrm{e}^{Z_{6,t}})P_t\mathrm{e}^{-Z_{6,t}} - \frac{\sigma_2^2}{2})\mathrm{d}t + \sigma_2\mathrm{d}W_{6,t}, \tag{4.18}$$

while the rest of the system is the same as in Eq. (4.12). Conveniently, the new system is successfully estimated by `ctsmr` and gives the following parameter estimates:

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
6.89	2.112	9.633	1.170	1.736	0.087	0.010	0.1	0.049	0.1

Again, A has increased compared to estimates in the previous steps, likely due to the updated noise structure. The visual appearance (Fig. 4.20) is similar to that of step 9 (Fig. 4.14) as expected, but this time all parameters have been estimated together in the proper way.

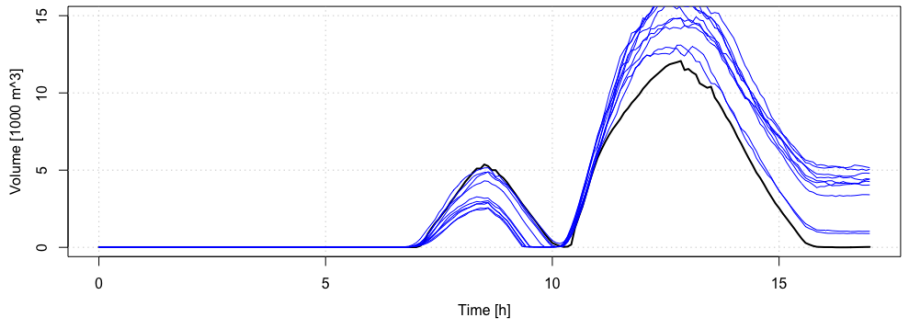

Figure 4.20: 10 realizations of event no. 7 by the CTSM estimated in step 11 (blue), compared with the observed volume (black).

4.2.12 Step 12 - fitting on multiple rainfall events

With step 11 every aspect of the CTSM structure has been resolved, and the only remaining task is to estimate it based on more than just one rainfall event. This is done in order to achieve a robust model that can handle as many kinds of rainfall events as possible. We decide that a set of six events (no. 3, 7, 8, 10, 13 and 17, see Appendix A) provides a good span of the variety of rainfall events that can occur.

Estimation based on multiple datasets is very straightforward with the existing estimation method established. All one-step predictions across all events are assumed to be independent, and hence the negative log-likelihood given the combined dataset, $\ell(\theta|\mathcal{Y}_3, \mathcal{Y}_7, \mathcal{Y}_8, \mathcal{Y}_{10}, \mathcal{Y}_{13}, \mathcal{Y}_{17})$, is simply equal to the sum of the individual negative log-likehoods given each of the corresponding datasets:

$$\ell(\theta|\mathcal{Y}_3, \mathcal{Y}_7, \mathcal{Y}_8, \mathcal{Y}_{10}, \mathcal{Y}_{13}, \mathcal{Y}_{17}) = \ell(\theta|\mathcal{Y}_3) + \cdots + \ell(\theta|\mathcal{Y}_{17}). \tag{4.19}$$

A graphical representation of the simultaneous fitting to the six datasets can be found in the Supplementary Material of Paper A (Paper A Fig. 6). Since this is the final step in the modelling process, the CTSM estimated on the combined dataset is the final model of this thesis. It is therefore appropriately presented and discussed in its own section (Section 4.3).

4.3 The final model

A nonlinear CTSM for the rainfall-response in the Damhus tunnel has now been identified. It has the following system equations,

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{2}{K_2}X_{3,t} \\ \frac{2}{K_2}X_{3,t} - (\frac{2}{K_2} + q(X_4)\frac{1}{K_0})X_{4,t} \\ q(X_4)\frac{1}{K_0}X_{4,t} - \frac{1}{K_3}X_{5,t} \\ \frac{1}{K_3}X_{5,t} - q_P(X_{6,t})P_t \end{pmatrix}\mathrm{d}t + \begin{pmatrix} \sigma_1 X_{1,t}\mathrm{d}W_{1,t} \\ \sigma_2 X_{2,t}\mathrm{d}W_{2,t} \\ \sigma_2 X_{3,t}\mathrm{d}W_{3,t} \\ \sigma_2 X_{4,t}\mathrm{d}W_{4,t} \\ \sigma_2 X_{5,t}\mathrm{d}W_{5,t} \\ \sigma_2 X_{6,t}\mathrm{d}W_{6,t} \end{pmatrix}, \quad (4.20)$$

and the observation equation,

$$Y_k = X_{6,t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e^2), \quad (4.21)$$

where $t_1, ..., t_N$ are time points equally spaced with 5 minutes in between, i.e. the sampling rate used for parameter estimation is 5 minutes. The parameter estimates are as follows:

A	α	β	K_1	K_2	K_3	K_0	σ_1	σ_2	σ_ε
6.101	5.356	5.627	1.386	1.814	0.074	0.602	0.1	0.067	0.1

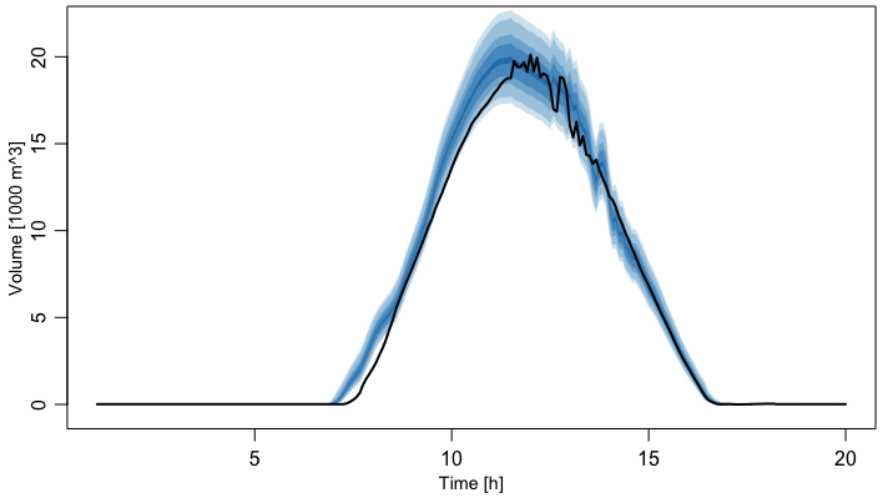

Figure 4.21: A series of probabilistic 1-hour forecasts on event no. 11 issued by the final model (Eq. (4.20)). The true observations are shown in black, and the 10%-, 50%-, 70%-, 90%- and 95%-quantiles of the forecasts are shown in nuances of blue.

After the parameters have been estimated, the model is ready to use for forecasting. Note that even though the sampling rate used for identification is 5 minutes, the model can be used for forecasting at any desired horizon. This property follows from the fact that the SDE-model provides a well-covering description of the system dynamics. For instance, as discussed in Paper A, 1-hour horizons are of interest, both because the overall retention time of the system is at the magnitude of a few hours, but also because if the model would later be integrated in a control scheme which optimizes pump usage around the intraday power market, then 1-hour forecasts would be needed. An example of the forecasting capabilities of the identified model is shown in Fig. 4.21. This graph displays a series of probabilistic 1-hour forecasts of event no. 11 (see also Fig. A.2).

Furthermore, Fig. 4.22 shows the reconstructed states for event no. 11. This reveals how the water flows between the six states. For example, it is clearly seen how the overflow crest prevents the water from flowing from X_4 to X_5 until the threshold of $X_4 = \beta = 5.627$ is met. It is also seen that the water does not spend much time in X_5 but discharges to X_6 very quickly, which is a consequence of K_3 being very small.

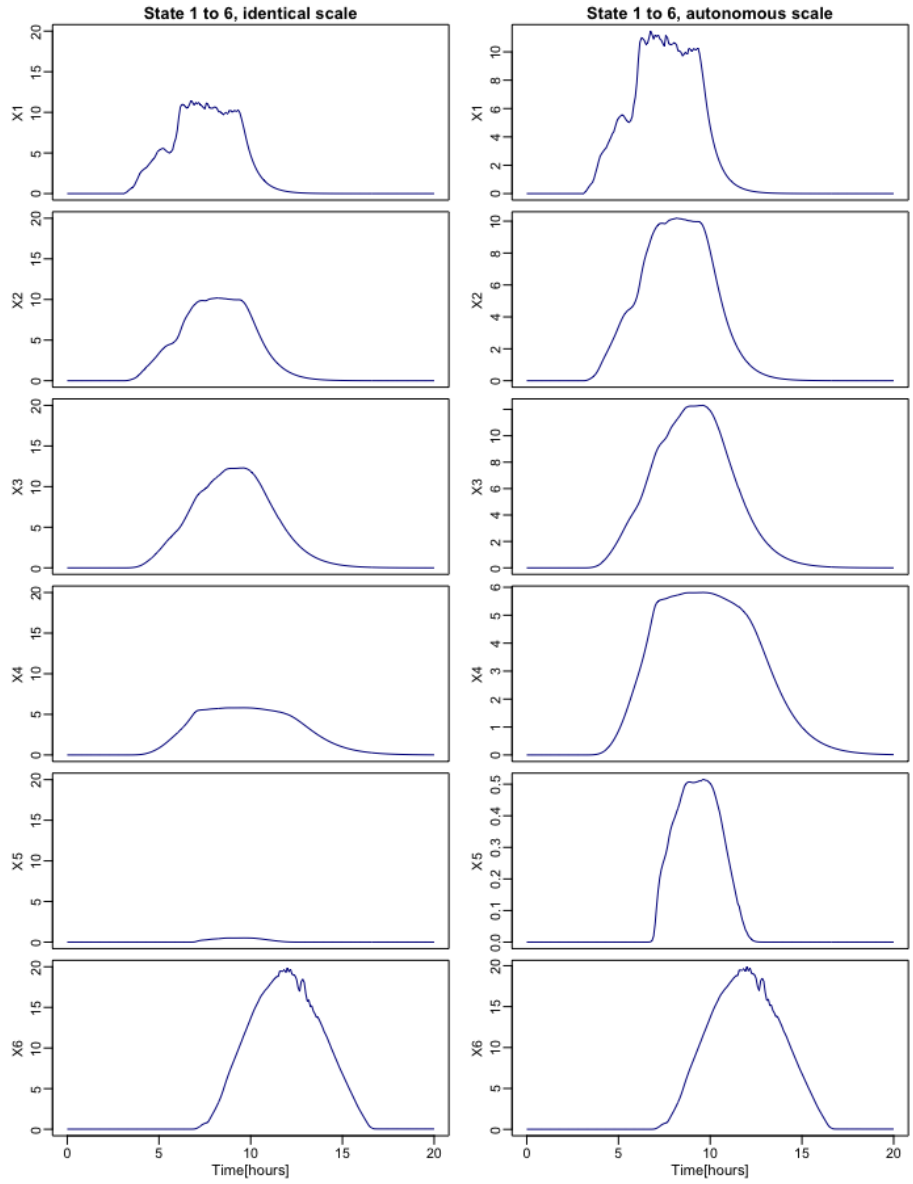

Figure 4.22: Reconstructed states for event 11 in units of 1000 m^3. The states from top to bottom are X_1, X_2, X_3, X_4, X_5 and X_6, respectively. The left and right columns show identical values, but the former has consistent scaling to enable comparison between states, while the latter is scaled to each state for visibility of the shape of the evolution of each individual state.

4.3.1 Summary of the model estimation

The final model was identified over the course of 12 steps, as explained in detail in Section 4.2. The modelling progression is summarized in Table 4.1. The table reports the evolution of the two most interesting physical quantities: the effective catchment area (A) and the overall average retention time ($\sum K$). It is seen that the choice of model structure has a big influence on A, which triples in magnitude from step 1 to step 12. The biggest changes to $\sum K$ happen when the overflow crest is introduced in step 3, and when the model is fitted to multiple datasets in step 12. Furthermore, the negative log-likelihood, ℓ, as well as the Bayesian Information Criterion (BIC) are reported. The latter can be used for heuristic comparison of model quality, under the assumption that the exact same response data were used to estimate the compared models (Neath and Cavanaugh, 2012). This assumption breaks when the training data is changed, hence in step 7 and in step 12. Generally it is seen that within comparable models, the BIC decreases over the steps, indicating a steadily improving model.

Step	A	$\sum K$	ℓ	BIC	Keynote
1	1.965	1.944	-102.228	-149.036	Simple linear CTSM
2	1.974	1.946	-100.979	-125.755	More time constants
3	2.672	2.572	-30.750	14.703	Crest function introduced
4	2.672	2.572	-30.750	-6.080	Reduce parameter space
5	2.676	2.516	-38.980	-8.684	Settle on 4 time constants
6	2.729	2.413	-78.585	-87.894	Settle on 6 system states
7	2.425	2.300	12.500	78.181	5-minute resolution
8	4.496	3.136	9.406	71.993	Combined sewer flow
9	NA	NA	NA	NA	State-dependent diffusion
10	NA	NA	NA	NA	Lamperti transform
11	6.890	3.003	-14.604	23.973	Attenuate pumping signal
12	6.101	3.876	83.833	220.847	Fit on multiple datasets

Table 4.1: Summary table of the step-by-step development of the CTSM in Eq. (4.20). The BIC column is highlighted in green, where the nuances indicate which models are fitted to the same data and hence are comparable under the BIC. No parameters were estimated in step 9 nor 10.

The final model was estimated in its Lamperti transformed version with $Z_i = \log(X_i)$ for all i, i.e. with the system equations,

$$\mathrm{d}\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \end{pmatrix}_t = \begin{pmatrix} AU_t \mathrm{e}^{-Z_{1,t}} - \frac{2}{K_1} - \frac{\sigma_1^2}{2} \\ \frac{2}{K_1}\mathrm{e}^{(Z_{1,t}-Z_{2,t})} - \frac{2}{K_1} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_1}\mathrm{e}^{(Z_{2,t}-Z_{3,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_2}\mathrm{e}^{(Z_{3,t}-Z_{4,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} - \frac{1}{K_0}q(\mathrm{e}^{Z_{4,t}}) \\ \frac{1}{K_0}q(\mathrm{e}^{Z_{4,t}})\mathrm{e}^{(Z_{4,t}-Z_{5,t})} - \frac{1}{K_3} - \frac{\sigma_2^2}{2} \\ \frac{1}{K_3}\mathrm{e}^{(Z_{5,t}-Z_{6,t})} - q_P(\mathrm{e}^{-Z_{6,t}})P_t\mathrm{e}^{-Z_{6,t}} - \frac{\sigma_2^2}{2} \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 \mathrm{d}W_{1,t} \\ \sigma_2 \mathrm{d}W_{2,t} \\ \sigma_2 \mathrm{d}W_{3,t} \\ \sigma_2 \mathrm{d}W_{4,t} \\ \sigma_2 \mathrm{d}W_{5,t} \\ \sigma_2 \mathrm{d}W_{6,t} \end{pmatrix}. \tag{4.22}$$

The parameters were estimated in the logarithmic domain and the identified optimum is nicely located in the interior of the log-parameter space, except in the cases of σ_1 and σ_ε which are both on the boundary. This is all visualized in Fig. 4.23 in terms of the profile negative log-likelihoods (Murphy and Van der Vaart, 2000) of each of the 10 parameters. It is here indicated that the optimzation routine wants to push σ_ε to lower values and σ_1 to higher values, thus trying to shift the relationship between system noise and observation noise. This is a typical behaviour from the CTSM optimization framework, and while one could argue that the boundaries should then be expanded, that is not compelling seen from a modeller's perspective.

The reason for this behaviour is that a very low observation noise will cause the filtered one-step predictions to stay very close to the observations, and thus make it much easier to achieve a low negative log-likelihood value. However, having very low observation noise while pushing the system noise up, is also equivalent to neglecting the importance of a good system description and solely relying on very precise observations. This will become a problem as soon as precise observations are not available and will result in horrendous forecasts. Therefore, it is necessary to impose a reasonable lower bound on the observation noise, such that the optimization routine is forced to find good physical parameter estimates on its crusade to minimize the negative log-likelihood. At the same time, the system noise should not be allowed to completely explode either, because forecasting with a too large system noise is equivalent to "anything can happen". Such a forecasting model is of no value to anyone, and therefore, the system noise is bounded as well.

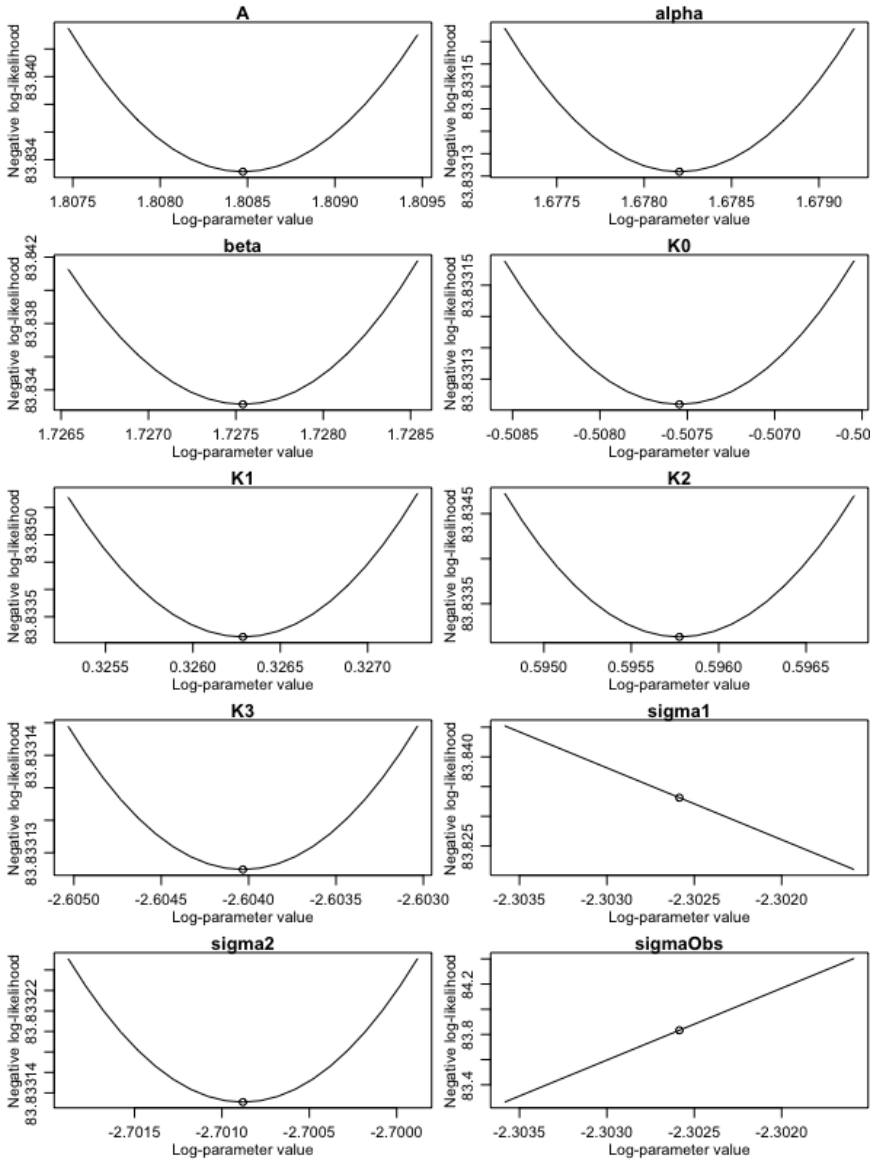

Figure 4.23: Profile negative log-likelihoods of the ten parameters in the final model.

Chapter 5

Forecast evaluation

Recall, the purpose of building a grey-box model for the Damhus case study (in this chapter referred to as "the SDE-model") was to be able to forecast the rainfall-response in the Damhus tunnel. Having completed this task, it is natural to ask: how well does the model perform in terms of its forecasting capabilities? Some *forecast evaluation* is appropriate in response to this.

Forecast evaluation is a surprisingly broad topic in the sense that a wide range of methods exist. In practice, for any forecasting problem, the methods of evaluation should be chosen with reason. Every evaluation method comes with some advantages and disadvantages, and these will greatly influence the extent to which said method evaluate the important aspects of the forecast in concern.

As previously stated, the forecasts issued by the SDE-model built in Chapter 4 are inherently probabilistic and temporally multivariate. Indeed, Paper B constitutes an introduction to multivariate probabilistic forecast evaluation and hence, the framework outlined in this work can be applied to the forecasts from the Damhus case study. The case studies in Paper B are focusing on wind power, so applying the framework to a different topic, namely stormwater forecasting, goes to demonstrate its strong, general applicability.

This chapter is structued as follows: Section 5.1 gives a brief introduction to the area of forecast evaluation and specifies what the aim is for evaluation in the context of the probabilistic stormwater forecasts. Section 5.2 introduces the actual methods used for the task, i.e. the so-called scoring rules, and finally, these scoring rules are applied to the stormwater forecasts issued by the SDE-model and compared with benchmark forecasts from a generic ARIMA-model (Madsen, 2007) in Section 5.3. All in all, Chapter 5 essentially demonstrates how the probabilistic forecast evaluation framework explained in Paper B can be applied to a new case, in this case forecasts of stormwater in the Damhus tunnel.

5.1 Forecast evaluation in the context of probabilistic stormwater forecasting

The following is based on the introduction from Paper B. Consider some arbitrary real process that is subject to forecasting. When the process is realized and observed, the quality of the forecasts can be assessed with respect to the observations. This practice is denoted forecast evaluation. The evaluation is usually performed in terms of a metric called a scoring rule (quantitative evaluation that allows for comparison with competing forecasts) or a statistical test (qualitative evaluation that allows for simple acceptance or rejection of the forecast).

For example, if the process is Gaussian, then the full distribution of the forecast is characterized by only its mean and variance. In that case, quantitative evaluation can reliably be performed by applying the usual root mean squared error (RMSE), while qualitative evaluation can be done by considering whether a sufficient share of the observations fall inside certain prediction intervals of the forecasts.

However, if the process is not Gaussian, it can no longer be assumed that the forecast distributions are symmetrical. Then the RMSE and the simple prediction intervals will be more and more misleading, the further from Gaussian the distributions are. Instead, it is better to find a generalized way to evaluate the entire distribution, or at least a part of it (for example a set of quantiles). An obvious (but not the only) choice is to consider the maximum likelihood of the forecast distribution with respect to the observation. Note that in the Gaussian case, the maximum likelihood and the RMSE will reach the same conclusions.

In the case of the stormwater forecasts produced by the SDE-model from Chapter 4, we are dealing with forecasts with the following properties:

- The forecasts are probabilistic. This follows because the evolution of the SDE-model is an evolution of a probability density
- The probabilistic forecasts are asymmetrical. This is a consequence of the state-dependent system noise.
- The $1, 2, \cdots$-step ahead forecasts are autocorrelated and hence multivariate.

Furthermore, keep in mind that the future ambition with the model is to integrate it with a model predictive control (MPC) scheme. If we only cared about saving energy and money on the pumping schedule, we might be able to just

consider point forecasts (e.g. the mean or median taken from the forecast densities) and achieve a satisfactory MPC scheme. However, the risk of flood must be taken into account, and this calls for an assessment of the uncertainty. This is a strong argument for striving for an evaluation of the full forecast distribution. Getting the autocorrelation right is a secondary objective but it is still important, because if it is wrong, then the uncertainty of the predicted accumulated water volumes will also be wrong, which in turn has impact on the assessment of the risk of flood.

Having pinpointed the important features of the stormwater forecasts to be that they are probabilistic and multivariate, we can now select the appropriate methods for forecast evaluation. Indeed, Paper B gives suggestions on how to do exactly that within the toolbox of scoring rules.

5.2 Scoring rules

A common way to evaluate a forecast is to apply a *scoring rule*. A scoring rule is a scalar function $S(G, y)$ of the forecast G and the corresponding observation y, and the returned value is called the *score* (Gneiting and Raftery, 2007). Typically, a scoring rule is defined such that lower scores reflect the better forecasts. Hence, if e.g. G_1 and G_2 are two competing forecasts of the same event, and if $S(G_1, y) < S(G_2, y)$, then for that specific event, G_1 is considered a better forecast than G_2 by S. For a robust evaluation, a suitable series of observations $\mathcal{Y} = (y_1, \cdots, y_N)$ should be considered, in which case the average score,

$$\bar{S}(G, \mathcal{Y}) = \frac{1}{N} \sum_{i=1}^{N} S(G, y_i) \tag{5.1}$$

can be used as the overall evaluation metric.

Many different scoring rules are available in the literature, each one with its own characteristics that may or may not be desirable for the evaluation problem at hand. In the case of the rainfall-response model, we need to evaluate multivariate probabilistic forecasts. Paper B finds that the most practical way to accomplish this is to split the evaluation problem into two parts. The first part concerns the calibration of the marginal distributions, and the second part concerns the correlation structure, and in turn, the multivariate aspect of the forecast. Hence, we introduce two scoring rules for those two evaluation parts.

5.2.1 The continuous ranked probability score (CRPS)

The continuous ranked probability score (CRPS) is an excellent method for evaluating univariate probabilistic forecast distributions (Matheson and Winkler, 1976). It considers the forecast in terms of its cumulative distributive function F, and is defined as follows:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(u) - \mathbb{I}(u \geq y))^2 \mathrm{d}u, \tag{5.2}$$

where $\mathbb{I}$ is the indicator function. The CRPS evaluates the entire forecast distribution, and has the neat property that the true forecast distribution will always, asymptotically, yield the best score. This property gives the assurance that if the CRPS finds that model X is clearly better than model Y given a suitable series of events, then the forecasts issued by model X *are* indeed more accurate than those issued by model Y.

Although the CRPS evaluates the full distribution, it is mostly sensitive to the calibration of the mean/median and overall shape, while it is not very sensitive to the tails of the distribution. In forecasting scenarios where differences in the tails are important, alternatives like the logarithmic score (Good, 1952) should be considered. Both the CRPS and the logarithmic score are thoroughly documented, investigated and compared in Paper B.

While the CRPS can be generalized to multivariate forecasts (Gneiting and Raftery, 2007), it is computationally slow for higher dimension and does not evaluate correlation structure very sensitively anyway, and is thus not worth the hassle to use for the multivariate aspect.

5.2.2 The variogram score (VarS)

For evaluation of the correlation structure, the relatively newly proposed variogram score (VarS) is very convenient (Scheuerer and Hamill, 2015). It is defined as follows,

$$\mathrm{VarS}_p(\phi, y) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} w_{ij} (|y_i - y_j|^p - E[|X_i - X_j|^p])^2, \tag{5.3}$$

where ϕ is the joint probability density function of the multivariate forecast, and X_i is a univariate random variable that follows the i'th marginal distribution

of ϕ (and analogously for X_j). The weights w_{ij} should be chosen in a way that reflects the relative importance of the individual correlations in the multivariate probabilistic forecast. p may be chosen freely as well, but $p = 0.5$ is strongly recommended for the nice sampling properties it yields (see Paper B).

The VarS is excellent at distinguishing between correctly and wrongly specified correlation structure. Furthermore, it is very fast to compute, even for high-dimensional forecasts. This is crucial, because it is intended for multivariate forecast evaluation in the first place. Its main drawback, however, is its complete lack of sensitivity to calibration of the mean. Consequently, two forecasts with the exact same correlation structure, but with different offsets, would receive the same score. Proper calibration of the forecasts are almost always important, and therefore the forecast evaluation should not be based on the VarS alone. The properties of the VarS are thoroughly investigated and demonstrated in Paper B.

Hence, when evaluating forecasts with VarS, it must always be kept in mind that only the multivariate aspect is properly evaluated, while the calibration of the forecast distribution is not. This is why we address the latter with the CRPS.

5.3 Applied stormwater forecast evaluation

It is now time to evaluate the probabilistic stormwater forecasts issued by the grey-box model developed in Chapter 4. As motivated in the previous section, we are going to split the evaluation into two parts, applying the CRPS and VarS, respectively.

5.3.1 Calibration of marginal forecast distribution with CRPS

In this subsection, the marginal forecast distributions are evaluated with the CRPS. First, let us specify which forecasts are subject to evaluation. We are both interested in how well the fitted SDE-model performs on different rainfall events but also on different horizons. We have got:

- 18 rainfall events
- Forecast horizons ranging from 5 minutes to many hours ahead, in intervals of 5 minutes.

Evaluating every possible combination would be a large experiment beyond the scope of this chapter. Instead, we evaluate:

- Every one of the 18 rainfall events in terms of 1-hour forecasts only (see Fig. 5.2).
- 7 different forecast horizons (5-min, 10-min, 30-min, 1-hour, 2-hour, 4-hour and 6-hour), but only on event no. 11 (see Fig. 5.3).

Doing the above should give an idea of both the forecasting capability on different events as well as on different horizons.

Furthermore, it is difficult to interpret the scores on their own in a meaningful way. Therefore, we introduce a benchmark model to generate competing forecasts, that can be held up against the forecasts issued by the SDE-model. For benchmarking, we consider the classical autoregressive integrated moving average (ARIMA) model of order (1,1,1), which has the form:

$$\nabla y_t = \phi \nabla y_{t-1} + \theta \varepsilon_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{5.4}$$

where ϕ, θ and σ are the parameters of the model (NB: these symbols are already used in this thesis, but are here reused briefly to state the ARIMA-model for the sake of consistency with the literature), ∇ is the difference operator, and the ε_t's are i.i.d. Order (1,1,1) refers to the number of lags considered for the observations, the order of differencing, and the number of lags considered for the error terms, respectively (Madsen, 2007). The ARIMA-model is a black-box model that requires nothing but a set of observations. This property makes it easy to apply, which is why it is a common choice for benchmarking.

We proceed to fit an ARIMA-model to the same 6 datasets which the SDE-model were fitted to. The estimated ARIMA-model is as follows:

$$\nabla y_t = 0.898 \nabla y_{t-1} - 0.385 \varepsilon_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, 0.0352^2), \tag{5.5}$$

Both coefficient estimates are significant with $p < 0.05$.

The model in Eq. (5.5) is much simpler than the SDE-model in the sense that it consists of only one equation and does not depend on any physical assumptions. It is therefore reasonable to expect a good SDE-model to have better forecasting performance than the ARIMA-model. Fig. 5.1 shows some examples of comparable forecasts issued by the two models. It is seen that both models produce accurate and sharp forecasts on very short horizons, but the SDE-model scales much better than the ARIMA-model, as the forecast uncertainty of the latter

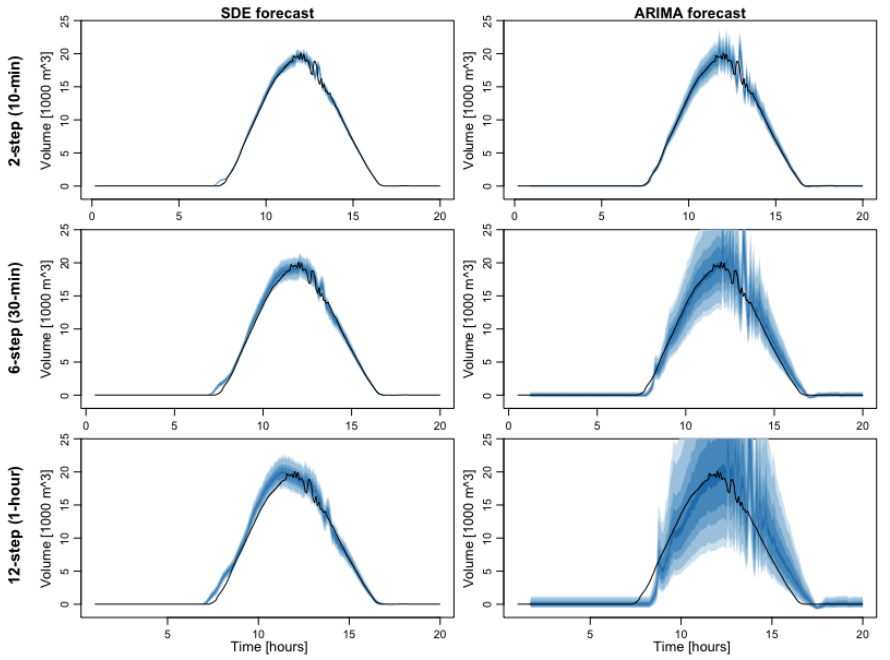

Figure 5.1: Probabilistic forecasts of rainfall-response in event no. 11 issued by the SDE-model (left column) and ARIMA-model (right column), respectively. Three different forecast horizons are shown for both models, namely 10-minute (top), 30-minute (middle) and 1-hour (bottom). The true observations are shown in black, and the 10%-, 50%-, 70%-, 90%- and 95%-quantiles of the forecasts are shown in nuances of blue.

quickly increases with horizon. 1-hour forecasts issued by both models on all 18 events are shown in Appendix A (Fig. A.4 and A.5).

It is also seen that the exact arrival time of the rainfall-response is not perfectly captured by the SDE-model. In some cases the response starts a bit too early, e.g. in event no. 10 and 11, while in other cases it starts too late, e.g. in event no. 13, 14 and 17. The reason is either a too simple model structure (e.g. the number of states or the parametrization of the overflow) or that more rainfall events are needed for identification. However, it is also evident that as soon as the first actual response is measured in the tunnel, the extended Kalman filter takes this discrepancy into account and adjusts the reconstructed states (re-estimates the amount of stormwater currently present throughout the system) such that any forecasts beyond that point in time are well-calibrated.

All the probabilistic forecasts defined by the restrictions above are then evaluated with CRPS. The result is displayed graphically in Fig. 5.2 and Fig. 5.3.

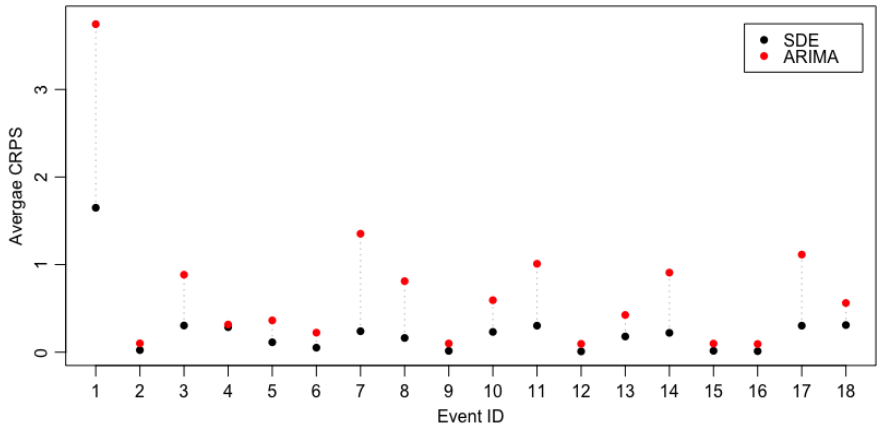

Figure 5.2: CRPS of 1-hour forecasts on events 1-18 issued by the SDE-model (black) and ARIMA-model(red), respectively.

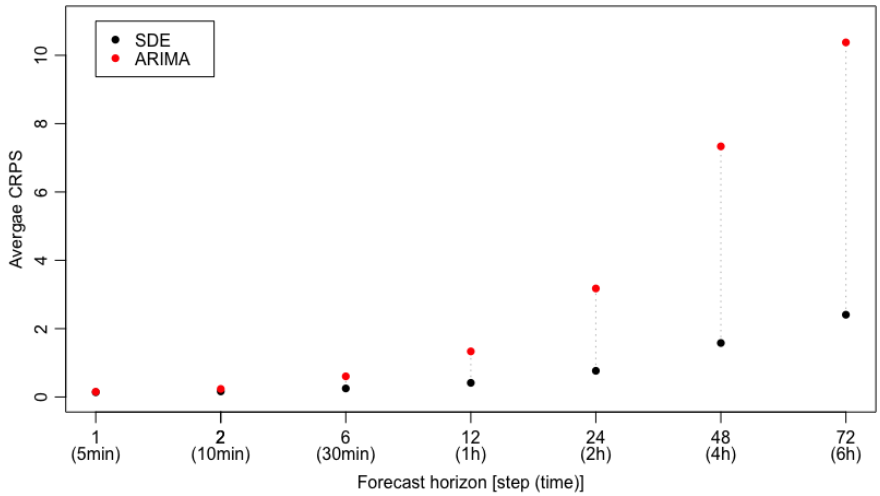

Figure 5.3: CRPS of forecasts on event no. 11 issued by the SDE-model (black) and ARIMA-model(red), respectively. Seven different forecast horizons are featured.

The scores behind the figures are listed in Appendix A (Table A.1 and A.2). The immediate conclusion is that the SDE-model always performs better than the ARIMA-model regardless of forecast horizon and event, although the difference in skill between the two models varies and both models are best at forecasting events with substantial amounts of rain (see e.g. event 11 in Fig. A.5). Neither are well-geared towards very small rain events (see e.g. event 9 in Fig. A.4).

Furthermore, as already indicated in Fig. 5.1, the performance of the SDE-model scales much better with horizon than the ARIMA-model.

5.3.2 Temporal correlation with VarS

In this part, the correlation structure of the forecasts is evaluated with the VarS. Again, it is in principle possible to do a mega-scale evaluation across different events, different multivariate dimension and different times at which the forecasts are issued. We will, however, just focus on providing a proof-of-concept example, which allows for being restrictive and concise.

We consider only *one* multivariate forecast instance, specified by:

- A 4-hour forecast horizon and hence a 48-variate probabilistic forecast.
- The 48-variate forecast is issued at $t = 100$ of event no. 11 (equivalent to 8 hours and 20 minutes into the event).

For benchmarking, we could use the ARIMA-model from the previous subsection, but it has proven to be horrible for increasing dimension, so we will instead

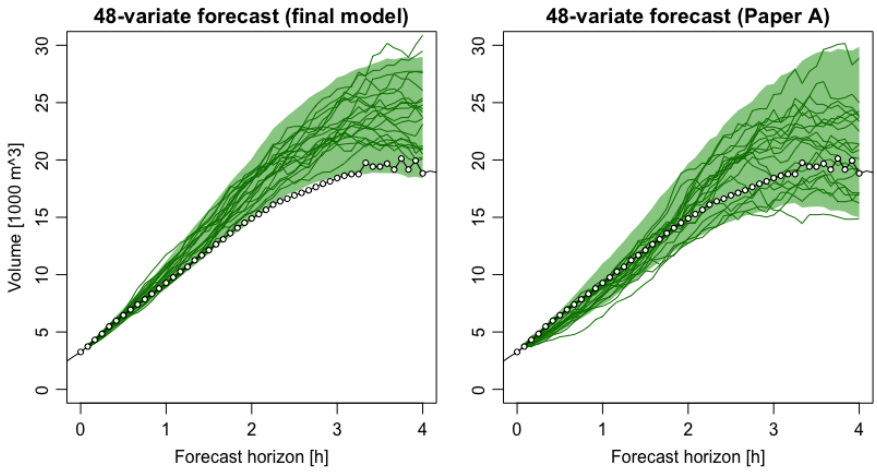

Figure 5.4: 48-variate (4-hour) probabilistic forecasts of rainfall-response in event no. 11 issued by the SDE-models developed in the thesis (left) and published in Paper A (right), respectively. 95%-prediction intervals are shown in pale green, with 20 random ensemble members shown in dark green. True observations are shown as points.

use the SDE-model from Paper A, which has slightly different parameter estimates than the SDE-model from Chapter 4. The two 48-variate probabilistic forecasts issued by those two models are shown in Fig. 5.4.

The VarS for the two models in this specific case are listed in Table 5.1. Weights $w_{ij} = 1/|i-j|$ are used in order to strengthen the importance of timely close correlations (see Eq. 5.3).

Model	VarS
Final SDE-model from Chapter 4	11.709
SDE-model published in Paper A	7.759

Table 5.1: Variogram scores for 48-variate forecasts of event no. 11 issued by the two competing SDE-models.

Recall that VarS only evaluates the multivariate aspect of a forecast, and in terms of this the model from Paper A is found to be performing the best. By inspection of Fig. 5.4, this is likely because the forecast realizations resemble the curved shape of the observation series better, i.e. the correlation structure is better captured by the model from Paper A. In contrast, the forecast realizations issued by the model from Chapter 4 diverge a bit from the observation series, and do not bend as much as they should if they were to be on par with the model from Paper A.

This concludes the proof-of-concept oriented evaluation example. To summarize, we first identified the important features of the stormwater forecasts in consideration, namely the uncertainty and autocorrelation. Consequently, we selected the two scoring rules that could appropriately evaluate these aspects, respectively the CRPS and the VarS. We then selected a small subset of stormwater forecasts generated by the SDE-model, to keep the example on a demonstrative level. Finally, an ARIMA-model was fitted to the stormwater observations and used for benchmarking where the CRPS and VarS of the forecasts of the two competing models were compared. If desired, this evaluation framework may be extended to any number of events, time points and forecast horizons, and other benchmarking models may be included. It is my hope that the reader now has a superficial understanding of how the evaluation of multivariate probabilistic forecasts like the ones produced in this PhD project can be approached.

Chapter 6

Concluding remarks

6.1 Revision of the objectives of the work

So what could be learned from this joyous journey? First of all, there are some conclusions which are directly tied to the objectives formulated in Chapter 1.

6.1.1 Contributions

- A non-linear SDE-based grey-box model of the stormwater response in the Damhus system has been developed. All of its parameters make physical sense, it is capable of producing ℓ-step probabilistic forecasts in a short time and the temporal correlation of the stormwater response is taken into account naturally.
- The step-by-step modelling section provides a read-this-first recipe for modellers who want to successfully develop grey-box models for dynamical systems in general.
- The question of how to evaluate multivariate probabilistic forecasts has been investigated. The studies show that calibration and correlation of forecasts, which are both very important, are almost impossible to evaluate simultaneously in practice. Therefore, the current best approach is to apply the univariate CRPS or the logarithmic score (LogS, see Paper B for details) to the marginal forecast densities and then evaluate the correlation with the VarS.

6.1.2 Suggestions for future work

- It would be good to test and re-estimate the SDE-model on rainfall forecasts rather than measured rainfall, to see what impact this difference would have on the bias and variance of the stormwater response forecasts.

- The developed SDE-model is promising with respect to a future model predictive control (MPC) application to the Damhus stormwater management system. If successful, the result of this would be a reduction of the energy consumption and financial expenses of the drain pumps associated with the Damhus tunnel.
- For multivariate probabilistic forecast evaluation, a well-defined guide on how to properly balance the weighting on calibration (with CRPS or LogS) and correlation (with VarS) is desired. In its current state, the suggested approach still requires heuristic decision making, when the CRPS/LogS and VarS disagree on which forecast is the best.

6.1.3 Elaboration on the revision of the objectives

The SDE-based grey-box model

Indeed, the developed grey-box model of the Damhus system was successful. It produced reasonable forecasts in a timely manner (see the documentation of run time in Paper A, Supplementary Material Section G) on different time horizons, with a reasonable covering of the forecast uncertainty.

The presence of an overflow structure in the modelled tunnel system meant that the system was heavily non-linear, which could not be handled by the usual linear methods. This was smoothly handled by adding a sigmoid function to resemble the overflow structure in the system description. The sigmoid function would ensure that a certain threshold of water had to be reached in the upstream states before any water could transition to the downstream tunnel states.

Using a fixed pumping signal caused the water volume in the model to sometimes attain negative values. This caused an unexpected problem, in the sense that the likelihood could no longer be evaluated, and hence the parameter estimation routine could not be completed. This was successfully solved, once again, with the help of a sigmoid function. This time around, it worked by attenuating the pumping signal whenever the present tunnel water got too close to 0. It seems that in any model where there is some sort of threshold, shift, delay or attenuation that has to be covered, the sigmoid option should always be considered.

Model predictive control

The above results are promising with respect to a future MPC integration into the Damhus stormwater management system. For such an MPC to be feasible, a handful of requirements must be met. First, it is required to have access to a model that can produce multi-step forecasts given a rainfall input forecast series and a pumping intensity sequence. Secondly, the forecasts must be produced fast enough to be ready for use by the MPC in due time. Both of these conditions are satisfied by the developed SDE-model (see the documentation of run time in Paper A, Supplementary Material Section G). For the design of the MPC, it is necessary to quantify the cost of pumping activity, both in terms of power consumption as well as the varying price on that power. The former should be possible to estimate from comparing the time series data of pumping volume and power which are available from HOFOR (the company responsible for operation of the Damhus tunnel). The latter is an integration of the power market into an MPC, which has been done in the literature from which a proper approach can probably be found and borrowed. It is also necessary to ensure that the pumping sequence suggested by the MPC is always feasible in practice. It must not change intensity at a rate faster than what is possible, and it must not exceed its maximum capacity. This information should be trivial to obtain from HOFOR. Finally, access to either rainfall forecasts or actual measurements in real-time with a sufficient updating frequency is needed. For example, the free API for the Danish Meteorological Institute data may be used to extract this information. More discussion of my thoughts on a future MPC application can be found in the discussion section of Paper A.

Forecast evaluation

With Paper B, I set out to shed light on the question of how to properly evaluate multivariate probabilistic forecasts. Here, 'properly' means to strive for forecasts that are correct in every aspect, namely in calibration and correlation. End users are mainly concerned with obtaining forecasts that maximizes value and minimizes losses, and in its current state this is not necessarily equivalent to finding the most correct forecasts. However, in my opinion this is a temporary phenomenon. If we can just learn how to use the information of a fully evaluated correct multivariate forecast distribution to construct appropriate cost functions to each relevant application, then that should be superior to simply letting forecast selection be a question of maximizing value among forecasts that lack full information about the distribution. How to achieve this change of practice will be a matter of future research.

With the mindset that the best way to evaluate forecasts are to evaluate the full multivariate distribution and select the most correct one, it is then concluded through the case studies of Paper B that the best approach is to evaluate the marginal forecast densities with the CRPS or LogS, and evaluate the correlation with the VarS. Calibration should take priority over correlation in almost any imaginable case. However, a problem arises, if one forecast clearly has the best VarS compared to its competitor, but a slightly worse CRPS. What decision should then be made? This question is currently at a heuristic state, and it is probably impossible to make a simple relative weighting of the two. The answer is rather to make well-covering guidelines that enable the user to make this decision in a consistent manner.

Regarding forecast evaluation of the stormwater forecasts issued by the grey-box model specifically, I reconfirmed what I already concluded in Paper B. Competing multivariate probabilistic forecasts can in practice be evaluated by applying the CRPS to the marginal distributions and then cover the correlation structure with the VarS. These conclusions are reliable because they are consistent with how said forecasts compared to each other graphically. I believe there is a lot of application potential in the concerned evaluation methods, which can be explored with additional relevant case studies in the future.

6.2 Lessons learned about modelling in practice

Secondly, there are some conclusions belonging to the 'fun challenges'-category. While it is not something one would report as a result in a scientific paper, it is certainly the kind of stuff that makes you a more skilled modeller when you have been exposed to - and overcome it. The issues with the likelihood not evaluating and the optimizer never converging led to several useful lessons learned:

1) If the likelihood fails to evaluate at some point, it is worth investigating exactly when it is feasible and when it is not. This meant the difference between being stuck indefinitely and finishing the development of the grey-box model and publish it in a journal paper.

2) Graphical inspection is extremely important for troubleshooting. This is probably well-known to every soul on the planet, but rest assured that I had to be reminded of this a couple of times during this adventure. Those two times (3+4) were the following:

3) Even after acknowledging lesson no. 1, it was very tricky to figure out why that likelihood function could not evaluate. I only made the realizing break-

through, when I made a graphical inspection of the temporary one-step predictions produced by the currently guessed model. This was when it became clear that the water could never be allowed to go into the negative. I had known for a while that ideally this should not happen, but I did not know that the restriction had to be rigorous to the point of no half measures. Even though I did not yet understand the underlying mathematical reasons for this, at the time, it was enough of a realization to propose a solution and apply it with a perfect result almost immediately.

4) This is something I have decided not to spend paper-space on in the core thesis. But I shall not refrain from reporting it in the concluding remarks. Even with a robust likelihood function that always would evaluate, it was often difficult to get the optimization routine to converge to a physically meaningful parameter estimate. Getting a consistent estimate given slightly different starting guesses or constraints were even more difficult. During optimization attempts, I would always be monitoring the parameter jumps in a live-updating table. Many times the model would seem to be converging for a while, but then it would start moving out on an endless tangent towards some ridiculous boundary point. And that would never end well. Making useful plots for graphical assessment of models with 10 or more parameters was no easy task, and for a long time I did not have any good ideas on how to graphically inspect this issue, aside from constructing a ton of time-consuming profile likelihoods and tables. However, one day when I was trying to optimize over the six training datasets, I got the idea of adding a live-updating graphical view of all six datasets along with the current model fit. I would describe this as a minor miracle. It led to the realization that the optimizing algorithm actually did find a good fit across all of the datasets, within a relatively small number of steps. After that point, most of the iterations would be minor improvements with diminishing returns, before moving in the famous endless no-prospect direction. The critical breakthrough was that I could suddenly make an informed decision on when to stop the optimization algorithm and deem a parameter estimate to be good enough. This was of course not the end of the optimization quest, since just stopping the alogrithm prematurely would return an estimate that wasn't a local minimum. But ultimately, it opened the door to the track that would eventually allow me to find a set of parameter estimates worth publishing.

5) Having realized that I should not blindly trust the optimizing routine, discussions with knowledgeable colleagues began. This led to the understanding that the relationship between system noise and observation noise levels can greatly affect the optimization flow. If unrestricted, the optimizer has a tendency to let the observation noise surge to the bottom. With minimal observation noise, the Kalman filter will correct the state estimations very aggressively, leading to an almost perfect fit. The problem with this is that the prediction quality no longer depends on the physical system structure, which is what we are trying

to estimate, after all. It was clear that this was the reason why the model had such a hard time getting properly estimated. To solve this issue, I would split the optimization framework into two legs. The strategy was to first optimize the physical parameters, and then optimize the noise parameters. In the first leg, the system noise would be fixed to a small level to ensure some trust in the physical part of the CTSM. It worked, and the model would converge to a physically reasonable parameter set. In the second leg, all the physical parameters estimated in the first leg would be fixed, and all of the noise parameters were due to optimization. This two-legged approach was successful and I ended up with a working parameter estimate that I could vouch for, mathematically.

Finally, I will dare to conclude that one does not easily overcome modelling challenges. You really need to stand up again after every time you fall, no matter how many beatings you have to endure. However, by all means, do consult someone when you are out of ideas, because simply trying the same thing again which already failed several times, almost never leads to anything productive. I am really happy to have finalized this marathon, knowing that I have genuinely become a more experienced modeller than I was when I started.

This concludes the summary report of my PhD thesis.

Appendix A

Supplementary material

A.0.1 Equations

In Section 4.2.11, the differential equation

$$\frac{\mathrm{d}z}{\mathrm{d}t} = p \cdot \mathrm{e}^{-z} \tag{A.1}$$

is stated (Eq. (4.14)) with its solution $z(t)$ (Eq. (4.15)). The following elaborates on how the solution is found. Multiply by e^z on both sides to get,

$$\mathrm{e}^z \mathrm{d}z = p\mathrm{d}t. \tag{A.2}$$

Integrate on both sides:

$$\int_{z_0}^{z} \mathrm{e}^u \mathrm{d}u = \int_{t_0}^{t} p\mathrm{d}w \implies \mathrm{e}^z = p \cdot t + c, \tag{A.3}$$

where c is an integration constant. Take the logarithm on both sides to get:

$$z = \log(p \cdot t + c). \tag{A.4}$$

Set the initial condition $z(0) = z_0$, then $z_0 = \log(c) \implies c = \mathrm{e}^{z_0}$. Inserting this result into Eq. (A.4) yields the solution,

$$z = \log(p \cdot t + \mathrm{e}^{z_0}). \tag{A.5}$$

The Bayesian Information Criterion (BIC) used in Section 4.3 is defined as

$$\mathrm{BIC} = k \log n + 2\ell, \tag{A.6}$$

where k is the number of model parameters, n is the number of observations and ℓ is the negative log-likelihood as usual.

A.0.2 Visual overview of the Damhus case data

In the following, all 18 rainfall events featured in the Damhus case (see Section 4.2) are visualized with rainfall and rainfall-response shown. All events are displayed on the same scale, such that individual comparison is straightforward.

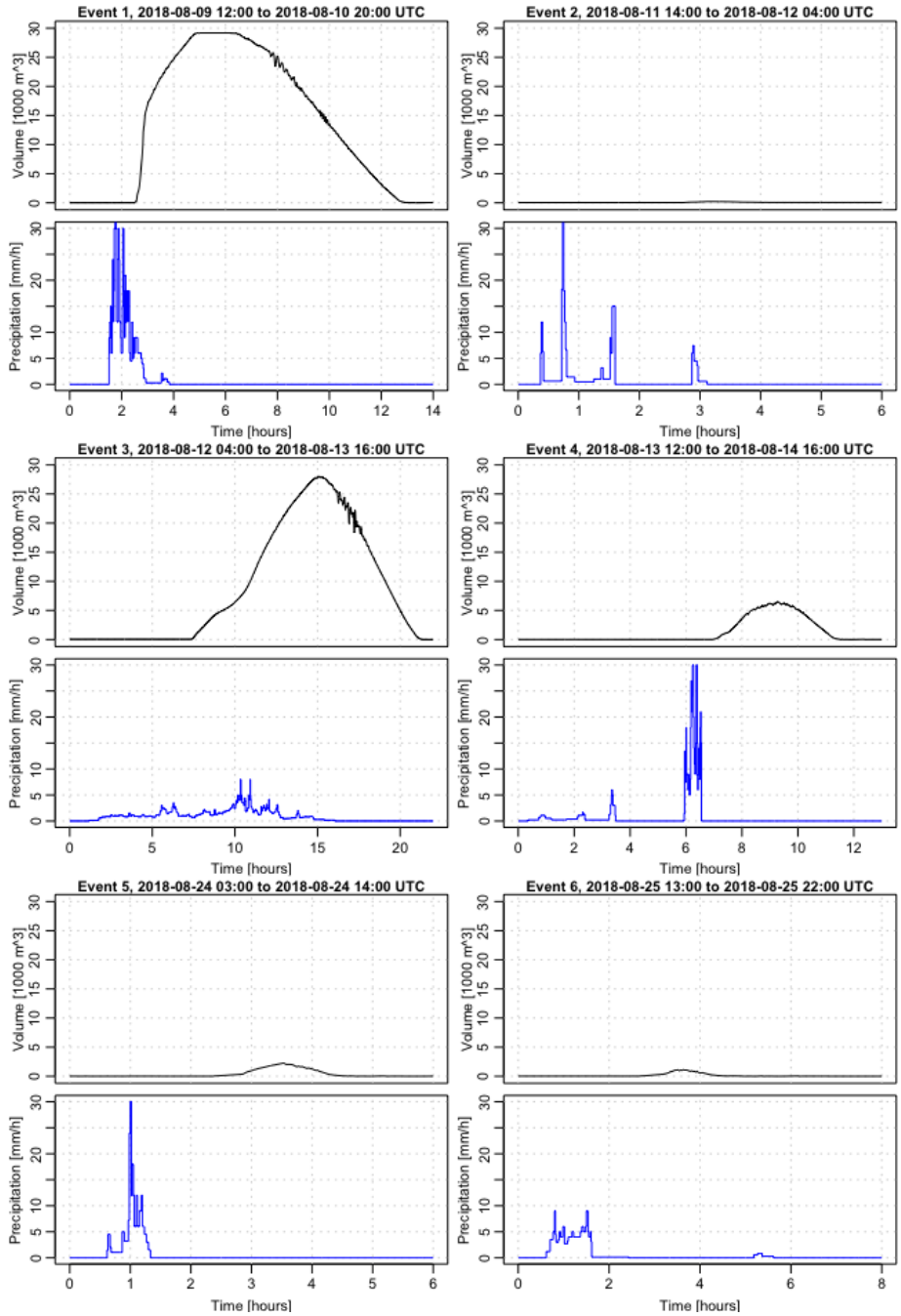

Figure A.1: Rainfall-response in rainfall events 1 to 6.

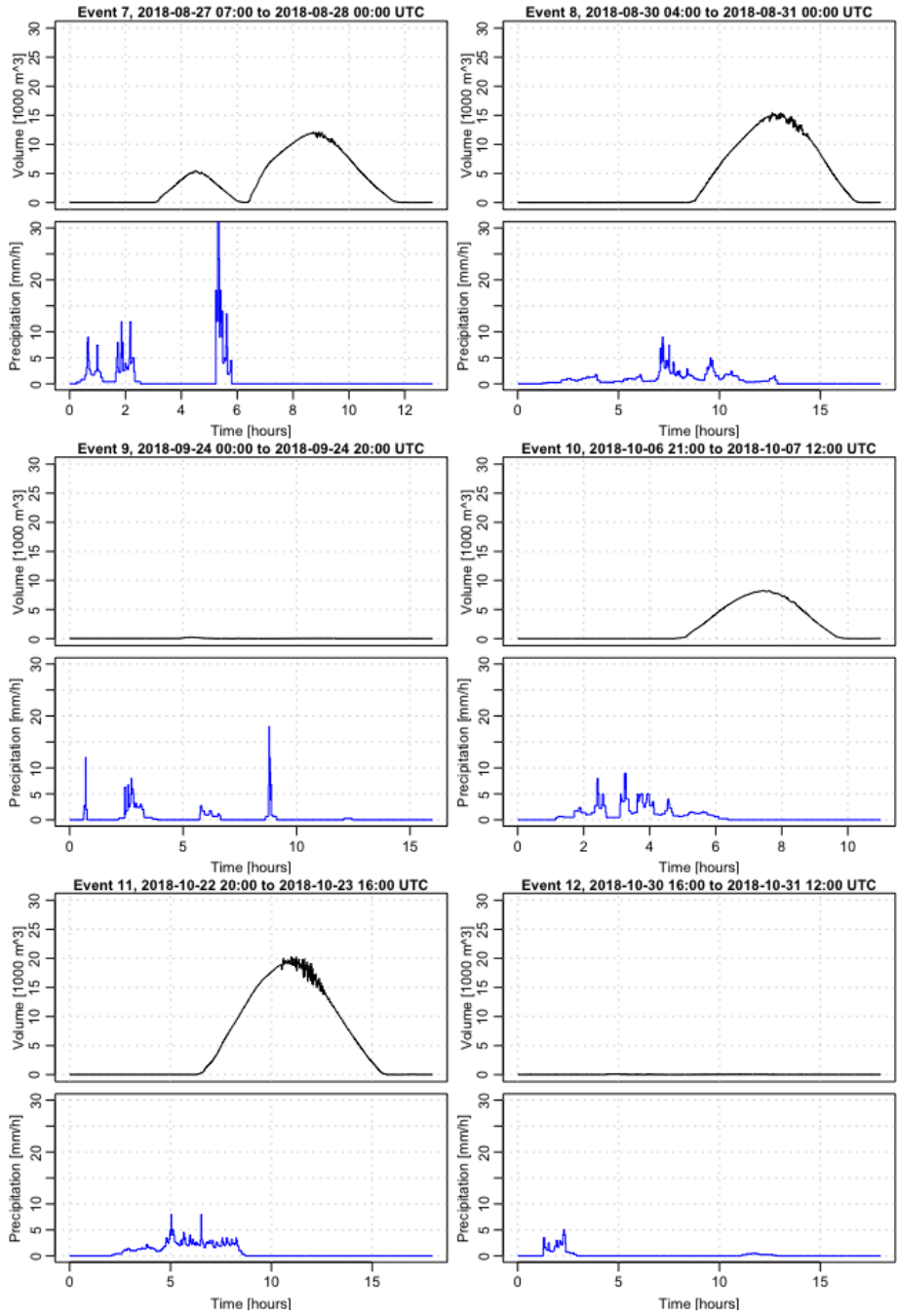

Figure A.2: Rainfall-response in rainfall events 7 to 12.

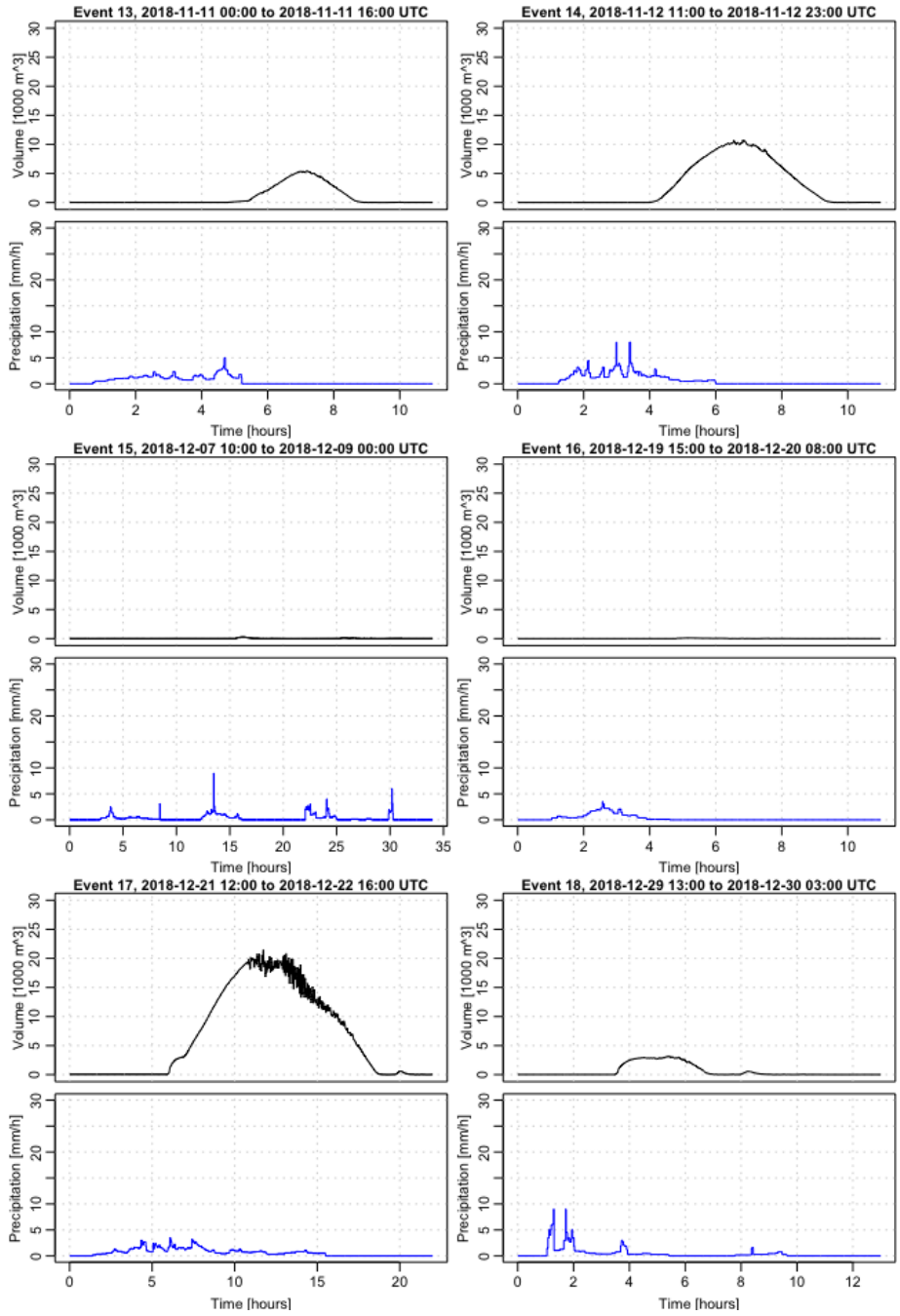

Figure A.3: Rainfall-response in rainfall events 13 to 18.

A.0.3 Probabilistic 1-hour forecasts of the 18 rainfall events

In the following, probabilistic 1-hour forecast series for all 18 rainfall events, issued by both the final SDE-model and the benchmark ARIMA-model (see Section 5.3.1) are shown as prediction intervals up to 95%. Every event is scaled with respect to itself, so careful attention needs to be payed to the axes when comparing the events visually.

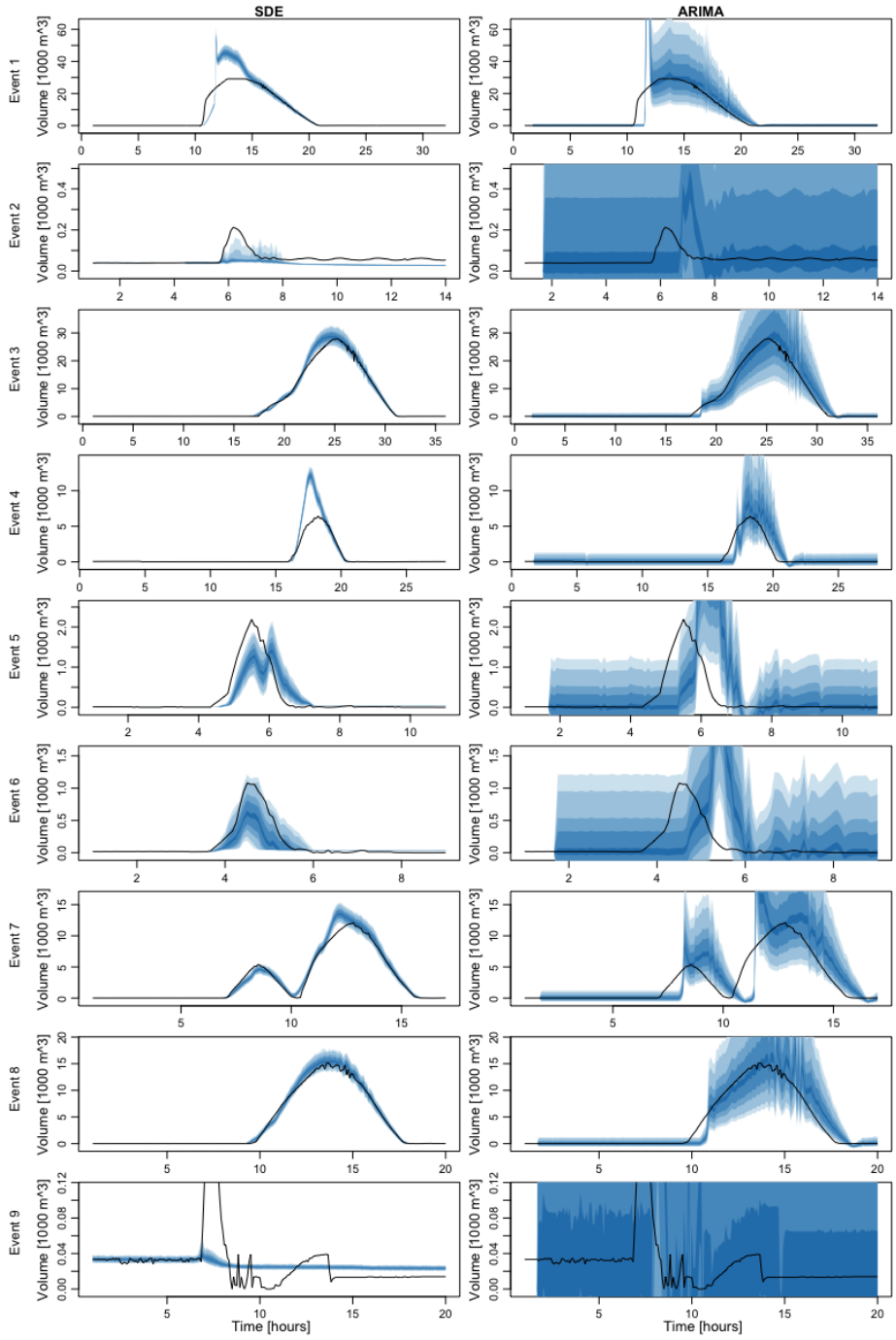

Figure A.4: Probabilistic forecast of rainfall-response in rainfall events 1 to 9 under the SDE and ARIMA models, respectively.

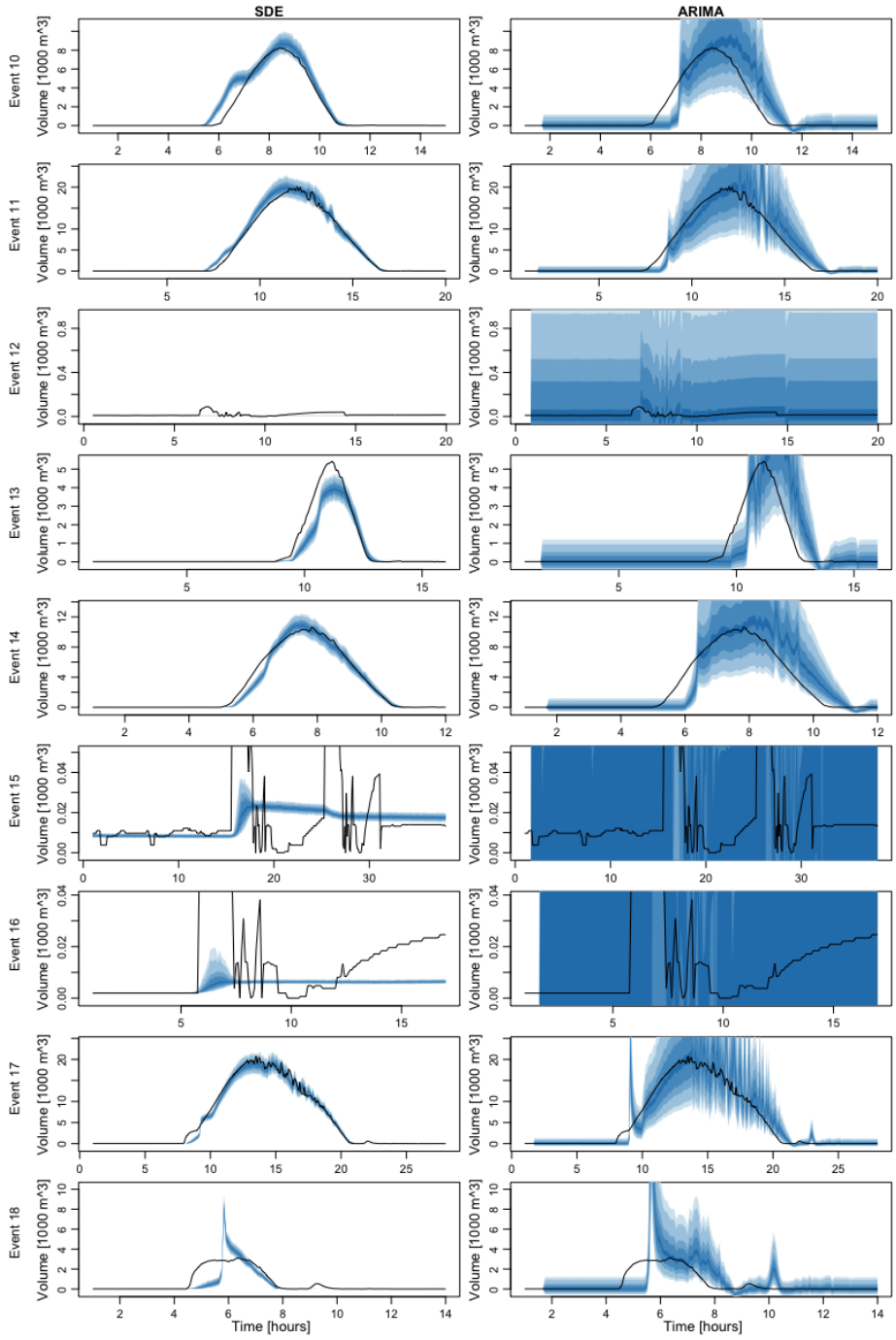

Figure A.5: Probabilistic forecast of rainfall-response in rainfall events 10 to 18 under the SDE and ARIMA models, respectively.

A.0.4 CRPS tables for forecast evaluation

Below follows the CRPS values used for display in Fig. 5.2 and 5.3 in Section 5.3.1.

Event	SDE	ARIMA
1	1.649	3.746
2	0.024	0.100
3	0.304	0.884
4	0.286	0.315
5	0.113	0.363
6	0.051	0.223
7	0.239	1.352
8	0.162	0.811
9	0.015	0.099
10	0.231	0.594
11	0.303	1.010
12	0.009	0.095
13	0.179	0.425
14	0.221	0.909
15	0.016	0.098
16	0.012	0.093
17	0.303	1.115
18	0.310	0.562

Table A.1: CRPS for probabilistic forecasts of all 18 events at 1-hour horizon (12-step forecast), for the final model (SDE) and the benchmark model (ARIMA) respectively. The scores are also displayed in Fig. 5.2

Step-ahead	Horizon	SDE	ARIMA
1	5-min	0.139	0.154
2	10-min	0.164	0.236
6	30-min	0.251	0.605
12	1-h	0.414	1.334
24	2-h	0.766	3.176
48	4-h	1.580	7.335
72	6-h	2.408	10.377

Table A.2: CRPS for probabilistic forecasts of event no. 11 at various horizons, for the final model (SDE) and the benchmark model (ARIMA) respectively. The scores are also displayed in Fig. 5.3

Appendix B

Publications

In the following, the published versions of my two journal papers are included. These are:

1. **Paper A**: Probabilistic forecasting of rainfall response in a Danish stormwater tunnel (*peer-reviewed, accepted and published in Journal of Hydrology*)
2. **Paper B**: An introduction to multivariate, probabilistic forecast evaluation (*peer-reviewed, accepted and published in Energy and AI*)

NB: The following errata have been identified and the journals have been contacted in an effort to get the official versions corrected.

Paper A: In both Eq. (11) and Eq. (12) the first term is stated as $\frac{N-1}{2}$ and the second term has a negative sign. The correct version should have $\frac{N}{2}$ as its first term, and the second term should have a positive sign.

Paper B: In Table 5, the LogS for the "True SDE" model is reported as 1.217. A negative sign is missing here, the correct value is -1.217.

Paper B: Fig. 7 reads: "*PIT histograms of a fixed times series under the true model (a), a mean-shifted, hence miscalibrated model (b), an underdispersed model (c) and an overdispersed model (d)*". It is (c) which is overdispersive and (d) which is underdispersive.

Journal of Hydrology 612 (2022) 127956

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

Probabilistic forecasting of rainfall response in a Danish stormwater tunnel

Mathias Blicher Bjerregård *, Jan Kloppenborg Møller, Niclas Brabrand Brok, Henrik Madsen, Lasse Engbo Christiansen

Technical University of Denmark – DTU Compute, Denmark

ARTICLE INFO

This manuscript was handled by Andras Bardossy, Editor-in-Chief, with the assistance of Zhenxing Zhang, Associate Editor

Keywords:
Stormwater forecasting
Non-linear stochastic differential equations
Linear reservoir models
Probabilistic forecasting
Urban drainage
Uncertainty evaluation

ABSTRACT

Sustainable urban drainage is an economically expensive necessity, partially due to the operation of water pumps. Reliable forecasting of stormwater response following a rainfall event has the potential to reduce those expenses, because it can be used in model predictive control schemes that optimize the energy consumption of pumps significantly better than the commonly applied real-time control systems. Urban drainage systems are traditionally designed around highly complex, deterministic models where an assessment of the uncertainty of the stormwater forecast is either absent or relies on computation–heavy simulations. With offset in a Danish stormwater tunnel, we propose a much faster, but reliable, non-linear continuous-discrete-time state-space model based on stochastic differential equations which can generate probabilistic forecasts that contain complete information about the distribution of uncertainty. We explain step-by-step how the model structure is built from simple physical assumptions, then how the parameters are estimated from maximum likelihood principles and finally we demonstrate the forecasting capabilities of the model. We believe this model would be well-suited for a subsequent model predictive control scheme.

1. Introduction

An accurate understanding of stormwater flow in sewer systems in cities is crucial for the prevention of floods caused by heavy rainfall events (Adams, 2000). If the dynamics of water flow in a system is well-described, it is possible to design the system in such a way that the expected abundance of flood is minimized to some level deemed acceptable, e.g. to a 10-year-event or 100-year-event (Schmitt et al., 2004). However, there are considerable financial costs associated with the management of stormwater. For example, the costs of stormwater management in California have been estimated to $700 million annually (EFC-Sacramento, 2020). A significant part of the costs can be attributed to the energy consumption for operation of drain pumps (Goldstein and Smith, 2002; Fecarotta et al., 2018), which is usually governed by real-time control schemes that act based on the current state of the system (Schütze et al., 2002). However, this practice is neither optimal with respect to energy consumption nor the price dynamics of the energy market. Instead, forecasting-based control schemes like model predictive control (MPC) (Morari and Lee, 1999) may be applied to reduce operational costs (Staden, 2011; Lund et al., 2018). Such schemes require reliable forecasting of the timing and scales of stormwater events, and hence there is a great potential in developing forecasting models for stormwater systems.

In practice, state-of-the-art modeling of urban drainage and sewer systems features deterministic methods implemented in software solutions like MIKE Urban (Wolfgang Rauch et al., 2002). In such frameworks a large set of partial differential equations describing the complex system of often thousands of sewer links, manholes and basins is solved numerically. However, this approach has a few obvious drawbacks. First, it is computationally expensive which is a problem in the case where updated forecasts are requested more frequently than model output can be computed (Hansen et al., 2014). Therefore, it might be beneficial to look for computationally cheaper options in the form of much simpler models. Secondly, a deterministic forecast does not carry any immediate information about the uncertainty. This is not ideal because a proper modelling of the uncertainty has the potential to dramatically reduce the risk of making a wrong decision (Hsu et al., 2012). Deterministic methods can handle this by feeding a range of perturbed inputs to the model and letting it propagate through time, thus generating a set of scenarios, to represent the uncertainty (Borup et al., 2015). However, this solution obviously suffers inherently from run time issues. Alternatively, stochastic differential equations (SDEs) can be used to model the uncertainty as well as the physical aspects of the system at the same time and thus constitute a method for probabilistic forecasting (Bechmann et al., 2000).

* Corresponding author.
E-mail address: matbb@dtu.dk (M.B. Bjerregård).

https://doi.org/10.1016/j.jhydrol.2022.127956
Received 2 March 2022; Received in revised form 8 May 2022; Accepted 17 May 2022
Available online 6 June 2022
0022-1694/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

Abbreviations

c.d.f	Cumulative distribution function
CRPS	Continuous ranked probability score
CTSM	Continuous-discrete-time state-space model
LRM	Linear reservoir model
MPC	Model predictive control
PIT	Probability integral transform
RMSE	Root mean square error
SDE	Stochastic differential equation

Mathematical notation

A	Effective catchment area
$f(\cdot)$	Drift function
$g(\cdot)$	Diffusion function
i	Reservoir index
k	Discrete time index
K	Generic time constant
K_0	Time constant for overflow
K_1	Time constant for ground surface
K_2	Time constant for combined sewer
K_3	Time constant for tunnel
l	Forecast horizon in steps
m	Ensemble size
N	Number of observations
P_t	Pumping intensity at time t
$q(\cdot)$	Sigmoid function for crest
$q_P(\cdot)$	Sigmoid function for pumping signal
t	Continuous time
U_t	Rainfall intensity at time t
$W_{i,t}$	Wiener process for reservoir i at time t
$X_{i,t}$	Water volume in conceptual reservoir i at time t
Y_k	Observation at time t_k as a random variable
y_k	Observation at time t_k as measured
$\widehat{y}_k$	Prediction of y_k
$Z_{i,t}$	Lamperti transform of X_i at time t
α	Shape parameter for sigmoid function
β	Threshold parameter for sigmoid function
θ	Vector of SDE-model parameters
σ_1	System noise parameter 1
σ_2	System noise parameter 2
σ_e	Observation noise

In this paper, we will consider the Damhus Tunnel system, a Danish stormwater tunnel not separated from the local wastewater sewer system (Jensen and Bering, 2017). Based on ideas of Breinholt et al. (2011), our key contribution is the development of a non-linear SDE-based state-space model for probabilistic forecasting of the stormwater response in the Damhus Tunnel, that can be used as a base for applied MPC in future studies. Section 2 describes materials and methods, Section 3 explains the modelling process, including choice of model structure, estimation, forecasting and forecast evaluation with respect to a prominent deterministic method. In Section 4, the results are presented, a discussion of the results follows in Section 5, and finally, Section 6 concludes.

2. Materials and methods

This case study features the Damhus tunnel in Copenhagen that accumulates stormwater from an approximately 47 km^2 large area called the Damhus catchment (for the future, simply referred to as the tunnel and the catchment, respectively). Previously, the drainage system of the catchment consisted solely of combined sewers, which insufficient maximum capacity occasionally led to very expensive floods, e.g. in July 2011. Therefore, the Damhus tunnel was constructed in 2017 to counter such events, offering an extra capacity of 29000 m^3. It is connected to the combined sewer system via an overflow structure named the Middle link. A simplified description of the full system in consideration is shown in Fig. 1 and further elaborated on in Section 3. This system has previously been used for development of a data assimilation scheme for urban drainage tunnels (Palmitessa et al., 2021).

2.1. Data

The data used for modelling includes measurements from the Damhus tunnel of water levels in meters above sea level (m(DVR)) and pump flows in m^3/min as well as rainfall measured in µm/s from two rain gauges installed at two different locations in the catchment, see Table 1 and Fig. 1. This data set spans five months of 2018 from August 1st to December 31st and contains 7 rainfall events where stormwater appeared in the tunnel. All the time series are available in 1-min resolution. Furthermore, we have access to a highly complex MIKE Urban model of the entire catchment that enables simulation of deterministic rainfall response for comparison. More details about the tunnel, the

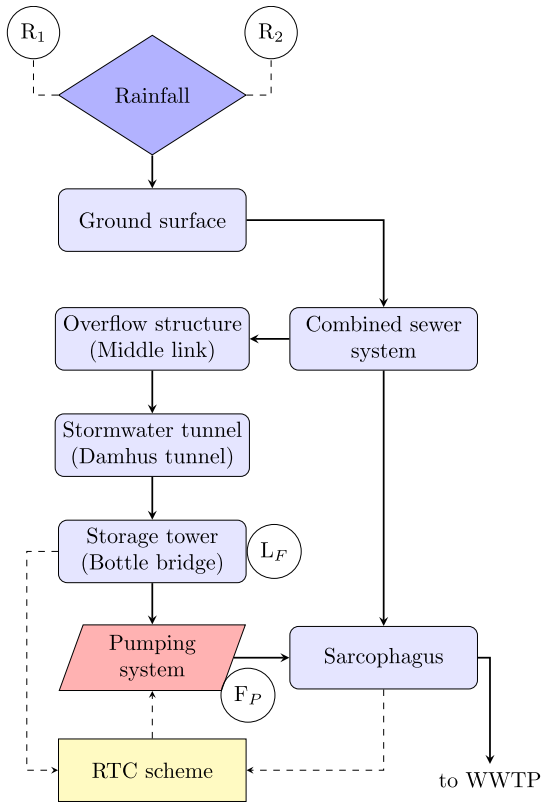

Fig. 1. Simplified diagram of the waste- and stormwater sewer system of the Damhus catchment. The presence of measurements used in this study are displayed as white circles.

Table 1
List of observations used in the modelling framework.

Sensor label	Description	Unit
R_1	Rainfall at location 1	µm/s
R_2	Rainfall at location 2	µm/s
F_P	Total pump flow at the Bottle bridge	m^3/min
L_F	Water level at the Bottle bridge	m(DVR)

rainfall events and the MIKE Urban model are included in Supplementary material.

2.2. Methods

The modelling framework consists of model structure selection, parameter estimation, forecasting, forecast evaluation and finally an assessment of the advantages and disadvantages associated with the proposed model. Throughout the case study we apply stochastic differential equations (SDEs) (Øksendal, 2003) as the building blocks for any model in consideration. A generic SDE describing the evolution of a state variable X_t can be formulated:

$$\mathrm{d}X_t = f(X_t, U_t, t)\mathrm{d}t + g(X_t, t)\mathrm{d}W_t, \tag{1}$$

where $f(\cdot)$ is called the drift term, $g(\cdot)$ is called the diffusion term, U_t is the vector of inputs and W_t is a standard Wiener process (Wiener, 1923). It is often necessary to model more than one state variable, in which case several SDEs are formulated and coupled to form a set of SDEs. The dynamics of the system is conveniently described in continuous time by the set of SDEs, whereas the data is almost always available in discrete time. Most often also only a subset of or a function $h(\cdot)$ of the states are measured, which gives rise the observation equation,

$$Y_k = h(X_{t_k}) + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e), \tag{2}$$

where Y_k is the observation at time t_k. Together, the set of SDEs and the observation equation constitute a continuous-discrete-time state-space model (Johansson et al., 1999), which we will commonly refer to as an SDE-model. We use the open software CTSM-R for estimation of the parameters of the model which is based on the log-likelihood and the extended Kalman filter (Juhl, 2020). We also apply the Lamperti transform to the system of SDEs prior to estimation, in order to be able to let $g(X_t, t)$ in Eq. (1) be state-dependent, because CTSM-R only accepts additive and state-independent system noise (Møller and Madsen, 2010).

For forecasting of Y at time $k + l$, i.e. $\widehat{Y}_{k+l|k}$, we use the estimated SDE-model to propagate forward in time with stochastic Euler simulation (Pardoux and Talay, 1985). This is repeated m times such that an m-dimensional ensemble is created. This ensemble is then considered a representation of the forecast distribution (Zhu, 2005), and is effectively a probabilistic forecast (Gneiting and Katzfuss, 2014). The performance of the probabilistic forecasts issued by the SDE-model is then evaluated against the corresponding deterministic forecasts generated from the MIKE Urban model.

3. Modelling

3.1. Physical drainage system overview

The full process of rainfall-runoff from precipitation to accumulated waste- and stormwater at the wastewater treatment plant (WWTP) can be divided into 6 steps:

1. The water hits the ground surface in the catchment.
2. It runs off along the ground surface to reach the combined sewer system.
3. It flows through the combined sewer system to reach the overflow structure (Middle link).
4. If the water in the overflow structure is above a certain crest level, it flows into the tunnel, otherwise it continues downstream through the combined sewer towards the Sarcophagus.
5. Water that has entered the tunnel flows downstream to the storage tower at the end (Bottle bridge).
6. The water is pumped from the downstream end of the tunnel to the Sarcophagus where it is merged with the sewage from the combined sewer (cf. step 4) and everything then flows towards the WWTP.

This breakdown of the process already reveals two important features of the system. Firstly, there are several different time constants reflecting how long it takes for the stormwater to discharge through the different phases. Secondly, the crest in step 4 introduces a non-linearity to the system, because it means that only sufficiently strong rain events will cause water to enter the tunnel, while minor rain events will not.

3.2. Water volume as response variable

Ultimately, the water level in the tunnel must be controlled such that flooding events are prevented. Hence the immediate idea would be to model the water level at any given time, an approach well demonstrated by Breinholt et al. (2011). However, water levels are not conserved in an intuitive way between different parts of the physical system, but water volumes are. Since the volume is proportional to the mass, choosing water volume as the response variable makes it straight-forward to base the model on mass balance equations, which have a very intuitive physical interpretation. From a time series of minutely water level measurements where the completely filled tunnel was being fully emptied, we know very precisely the relationship between the water level and volume in the tunnel, making transformation between these two domains easy, see Fig. 2. Hence, the water level can always be reconstructed whenever it is needed. Furthermore, all water volumes will be modelled in units of 1000 m^3 because estimation of the SDE-model tends to be easier when the order of magnitude of the numbers concerned is not too high.

3.3. Continuous-discrete-time state-space model

We shall formulate a continuous-discrete-time state-space model (Johansson et al., 1999) that can be estimated from the available data. First, consider an arbitrary sewer pipe. Let $X_{i,t}$ be a random variable that represents water volume at location i in the pipe at time t and let K be an associated time constant, i.e. the average time it takes the water to flow from one end of the pipe to the other. Then the flow may be described by a linear reservoir model (LRM) (Pedersen et al., 1980) of order n, where the pipe is represented by a series of n reservoirs with the water content of each reservoir being a state. The water then flows from one state to

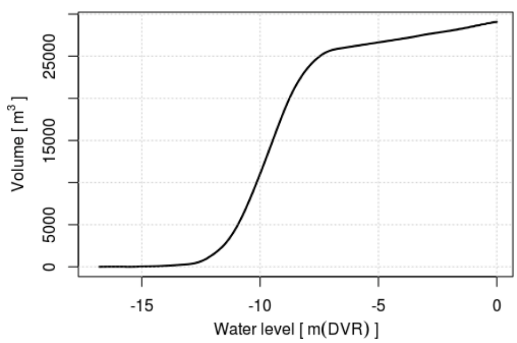

Fig. 2. The relationship between water level and water volume in the Damhus tunnel.

the next with the rate n/K, see Fig. 3. Ideally, the system consists of an infinite number of infinitely small reservoirs, but in practice only a limited number of reservoirs is needed. Because we are modelling water volumes, the dynamics of the i'th reservoir can now be described by a mass balance equation, typically formulated as

$$\mathrm{d}X_{i,t} = \left(\frac{n}{K}X_{i-1,t} - \frac{n}{K}X_{i,t}\right)\mathrm{d}t. \tag{3}$$

In order to model the entire Damhus system, we will consider the ground surface, the combined sewer and the tunnel as three different "pipes", each characterized by a time constant, K_1, K_2 and K_3 respectively, and use this idea to construct a state-space model consisting of three connected LRMs, each with 2 states, amounting to a total of 6 states. The decision on the number of states per LRM was based on preliminary studies. The time variable is considered in units of hours, and hence, the three time constants also have units of hours.

Regarding the input to the system, it is assumed that the amount of stormwater reaching the ground surface at time t is proportional to the rainfall intensity U_t with the proportionality constant being the effective area A, so the inflow is AU_t. This assumption was shown to be reliable by Breinholt et al. (2011). The mass balance for the first state X_1 of the LRM representing the ground surface thus becomes

$$\mathrm{d}X_{1,t} = \left(AU_t - \frac{2}{K_1}X_{1,t}\right)\mathrm{d}t. \tag{4}$$

Here, A and U_t are given in units of km^2 and mm/h, respectively, and hence, AU_t is in units of 1000 m^3/h as desired. U_t is taken as the average of the rainfall at the two measurement locations at time t, converted to mm/h, see Table 1.

The mass balances for X_2 and X_3 inherit the form in Eq. (3). At X_4, which represents the volume in the overflow structure, we address the non-linearity caused by the crest between the combined sewer and the tunnel. The discharge from X_4 should reflect that only if the amount of water exceeds some threshold corresponding to the crest level, water will start flowing rapidly into the tunnel. This can be accomplished by attributing a special time constant K_0 to the overflow, and multiplying the overflow-specific discharge $-\frac{1}{K_0}X_{4,t}$ with a sigmoid function,

$$q\left(x\right) = \frac{1}{1+\exp(-\alpha(x-\beta))}, \tag{5}$$

where α determines the sharpness, such that for $\alpha\to\infty$, $q(x)$ approaches a step function, and β is the threshold where the step occurs. Meanwhile, some discharge from X_4 downstream to the Sarcophagus will always take place, which is attributed to K_2. Hence, we get

$$\mathrm{d}X_{4,t} = \left(\frac{2}{K_2}X_{3,t} - \left(\frac{2}{K_2} + q\left(X_{4,t}\right)\frac{1}{K_0}\right)X_{4,t}\right)\mathrm{d}t. \tag{6}$$

The mass balance for X_5 which represents the upstream part of the tunnel, inherits the form in Eq. (3) except the inflow is multiplied by $\frac{1}{K_0}q\left(X_{4,t}\right)$,

$$\mathrm{d}X_{5,t} = \left(\frac{1}{K_0}q\left(X_{4,t}\right)X_{4,t} - \frac{1}{K_3}X_{5,t}\right)\mathrm{d}t. \tag{7}$$

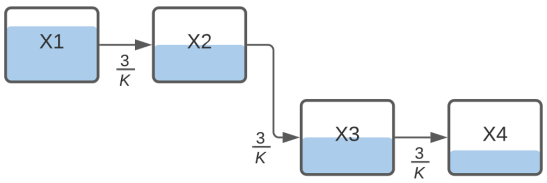

Fig. 3. A simple linear reservoir model with 4 reservoirs and one time constant K, hence the rate of discharge at each link is $3/K$.

The last state X_6 represents the water volume at the end of the tunnel. Here, there is no discharge and the water accumulates until it is pumped out or the tunnel is filled. The mass balance equation is

$$\mathrm{d}X_{6,t} = \left(\frac{1}{K_3}X_{5,t} - P_t\right)\mathrm{d}t, \tag{8}$$

where P_t is the pumping intensity at time t. This is an input signal, which depends on several factors in the real system. For this study, we use the pumping data at hand as a simplification, i.e. P_t is taken as the total pump flow at time t, converted to 1000 m^3/h, see Table 1. This has the awkward drawback of allowing the water volume to drop below 0. In order to ensure that such an occurrence never happens, P_t is multiplied by $q(X_{6,t})$ with $\alpha = 200$ and $\beta = 0.05$, cf. Eq. (5). We denote this function $q_P(x)$.

Finally, the diffusion terms in Eq. (1) must be selected. The simplest option is additive diffusion, i.e. $g_i(X_i, t) = \sigma_i$ for all i. However, Breinholt et al. (2011) demonstrated that state-proportional diffusion is the better option for flow modelling, i.e. $g_i(X_i, t) = \sigma_i X_i$. This causes the system noise to grow with larger volumes and converge to zero for smaller volumes. Consequently, no water volume state can attain any negative values, which keeps the system consistent with physics. Therefore, the latter option is chosen.

The diffusion constants, σ_i, may be modelled as unique for each i or with some of them being identical to one another. Here, the distinction is made based on physical assumptions about the sources of system noise. Firstly, it is assumed a part of the system noise is attributed to the rainfall input which enters the system in Eq. (4). For this reason, σ_1 is considered to be unique. Secondly, the remaining system noise is assumed to be attributed to random variation in the water flow which is not captured by the drift term of the model. We assume that this source of system noise is approximately the same throughout the entire drainage process and is hence only characterized by one diffusion constant, σ_2. Hence, the full system description becomes

$$\mathrm{d}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}_t = \begin{pmatrix} AU_t - \frac{2}{K_1}X_{1,t} \\ \frac{2}{K_1}X_{1,t} - \frac{2}{K_1}X_{2,t} \\ \frac{2}{K_1}X_{2,t} - \frac{2}{K_2}X_{3,t} \\ \frac{2}{K_2}X_{3,t} - \left(\frac{2}{K_2} + q\left(X_{4,t}\right)\frac{1}{K_0}\right)X_{4,t} \\ q\left(X_{4,t}\right)\frac{1}{K_0}X_{4,t} - \frac{1}{K_3}X_{5,t} \\ \frac{1}{K_3}X_{5,t} - P_t q_P\left(X_{6,t}\right) \end{pmatrix}\mathrm{d}t + \begin{pmatrix} \sigma_1 X_1 \mathrm{d}W_{1,t} \\ \sigma_2 X_2 \mathrm{d}W_{2,t} \\ \sigma_2 X_3 \mathrm{d}W_{3,t} \\ \sigma_2 X_4 \mathrm{d}W_{4,t} \\ \sigma_2 X_5 \mathrm{d}W_{5,t} \\ \sigma_2 X_6 \mathrm{d}W_{6,t} \end{pmatrix} \tag{9}$$

The observation equation simply relates the final state $X_{6,t}$ to the observed volume at the Bottle bridge,

$$Y_k = X_{6,t_k} + e_k, \quad e_k \sim \mathcal{N}(0, \sigma_e). \tag{10}$$

All the parameters of the model are summarized in Table 2 of Section 4.

3.4. Estimation

For estimation of parameters, we base our approach on the commonly applied minimization of the negative-log likelihood (Madsen and Thyregod, 2010):

$$\ell\left(\theta; y\right) = \frac{N-1}{2}\log\left(2\pi\right) - \frac{1}{2}\sum_{k=0}^{N-1}\log\left(\widehat{\sigma}^2_{k+1|k}\right) + \left(\frac{y_{k+1} - \widehat{y}_{k+1|k}}{\widehat{\sigma}_{k+1|k}}\right)^2, \tag{11}$$

where $\theta = (A, \alpha, \beta, K_0, K_1, K_2, K_3, \sigma_1, \sigma_2, \sigma_e)'$ is the vector of model pa-

rameters. First of all, we consider a 5-min resolution of the data set instead of 1-min resolution. Preliminary studies have shown that the resulting estimates do not change significantly and the computational time gain is huge. Secondly, instead of minimizing the regular log-likelihood, we make two suitable changes. The first change is attributed to the impact of small observations. While the data contain a lot of observations with values close to 0, we are primarily concerned about obtaining a model that can accurately predict the system load when water is present, and therefore, we want to put less emphasis on prediction of the former. This is accomplished by adding a variance contribution to $\widehat{\sigma}^2_{k+1|k}$ that has the largest effect on small observations:

$$\widetilde{\sigma}^2_{k+1|k} = \widehat{\sigma}^2_{k+1|k} + \left(a \cdot \widehat{y}_{k+1|k}\right)^b, \tag{12}$$

where a and b are both chosen to be 0.1. The second change aims to give special attention to an accurate modelling of the stormwater response time, i.e. the delay from rainfall to measurable water in the tunnel. During optimization, the regular log-likelihood may very well compromise w.r.t. this feature for the sake of a seemingly better fit overall, and therefore we add an extra penalty to Eq. (11) of the form $(y_{k+1} - \widehat{y}_{k+1|k})^2/(y_{k+1} + c)$, with $c = 0.01$, which penalizes a misspecified response time hard. The purpose of c is to avoid division by zero. Hence, the modified negative log-likelihood becomes:

$$\begin{aligned}\widetilde{\ell}(\theta; y) = \frac{N-1}{2}\log(2\pi) - \frac{1}{2}\sum_{k=0}^{N-1}\log\left(\widetilde{\sigma}^2_{k+1|k}\right) \\ + \frac{\left(y_{k+1} - \widehat{y}_{k+1|k}\right)^2}{\widetilde{\sigma}^2_{k+1|k}} + \frac{\left(y_{k+1} - \widehat{y}_{k+1|k}\right)^2}{y_{k+1} + c}.\end{aligned} \tag{13}$$

The total data set includes 7 rain events from 2018, cf. Section 2. For estimation, we select 6 events deemed sufficient to span the variation of the system, and one event for subsequent evaluation of the estimated model. Prior to estimation based on the 6 training events, we apply leave-one-out cross-validation (Cawley et al., 2003) to ensure approximately unbiased parameter estimates.

To obtain the 1-step predictions needed for the log-likelihood function, it is necessary to apply an extended Kalman filter (Brok et al., 2018). However, because the extended Kalman filter in CTSM-R only accepts additive and state-independent diffusion, the form in Eq. (9) can not be estimated. Instead, let $Z = (Z_1, \cdots, Z_6)'$ be a new multivariate random variable. We then apply a Lamperti Transform (Møller and Madsen, 2010) to the system, such that

$$Z_i = \log(X_i), \quad \forall i. \tag{14}$$

Hence we get a new system description of the form:

$$\mathrm{d}Z_t = \widetilde{f}\mathrm{d}t + \widetilde{g}\mathrm{d}W_t \tag{15}$$

with

$$\widetilde{f} = \begin{pmatrix} AU_t e^{-Z_{1,t}} - \frac{2}{K_1} - \frac{\sigma_1^2}{2} \\ \frac{2}{K_1}e^{(Z_{1,t}-Z_{2,t})} - \frac{2}{K_1} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_1}e^{(Z_{2,t}-Z_{3,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} \\ \frac{2}{K_2}e^{(Z_{3,t}-Z_{4,t})} - \frac{2}{K_2} - \frac{\sigma_2^2}{2} - \frac{1}{K_0}q\left(e^{Z_{4,t}}\right) \\ \frac{1}{K_0}q\left(e^{Z_{4,t}}\right)e^{(Z_{4,t}-Z_{5,t})} - \frac{1}{K_3} - \frac{\sigma_2^2}{2} \\ \frac{1}{K_3}e^{(Z_{5,t}-Z_{6,t})} - P_t q_P\left(e^{Z_{6,t}}\right)e^{-Z_{6,t}} - \frac{\sigma_2^2}{2} \end{pmatrix} \tag{16}$$

and $\widetilde{g}$ is analogous to the diffusion term in Eq. (9) except with no X_i. Thus, the diffusion of Eq. (15) is additive and allows for application of the extended Kalman filter. The physical parameters are invariant under this transformation, hence they can be estimated as in Eq. (16), and the fitted values of Z_i can be back-transformed to X_i by

$$X_i = e^{Z_i}, \quad \forall i. \tag{17}$$

A detailed calculation of the Lamperti transform can be found in the Supplementary material.

Finally, the modified negative log-likelihood in Eq. 13 is optimized using the `nlminb` function in R. Such optimization routines tend to converge more robustly if all the parameters are on a similar scale. For this reason, we let `nlminb` optimize over the log-transformed parameters. The parameter estimates in the original domain can thus be restored by taking the exponential of the result of the optimization. See the Supplementary material for a summary of the estimation framework in the form of R code.

3.5. Forecasting and forecast evaluation

The fitted SDE-model is now evaluated by letting it forecast the stormwater response in the test event. We shall use two different forecasting setups. The first setup features probabilistic forecasts with a moving l-step horizon, which is the setup we have in mind for an MPC application in future studies. The second setup is a forecast of the full scenario, given only the rainfall input and the initial conditions. The purpose of this is to compare the performance of the probabilistic SDE-model to the deterministic MIKE Urban model.

First, we cover the moving l-step forecasts. Based on the fitted SDE-model, using an extended Kalman filter we can reconstruct the full time series for all 6 states. These can then be used as initial conditions for forward Euler simulation of the model using the input series of the test set. We then save all the l-step forecasts generated this way, as a single series of l-step forecasts. By repeating that exercise m times, an ensemble of m members is obtained, from which predictive distributions can be extracted and used as probabilistic forecasts (Bjerregård et al., 2021).

Secondly, we would like to evaluate the SDE-model against the MIKE Urban-model. However, we can not directly compare with the moving l-step forecasts from the SDE-model, because MIKE Urban just simulates the full event based on rainfall input, and can not benefit from new observations or start the simulation in the middle of the event. Instead, to ensure the most fair comparison, we let the SDE-model forecast the full event with no extended Kalman filter updates, and hence it only depends on the rainfall input like MIKE Urban. Furthermore, we remove the pumping signal from the scenario, i.e. set $P_t = 0$ for all t, which is equivalent to forecasting the accumulated inflow of stormwater. This is due to the fact that the real pumping signal is not properly modelled in either model, so forecasting the instantaneous stormwater volume is not practical. Under these conditions, we generate a deterministic forecast series with MIKE Urban and a probabilistic forecast series with the SDE-model.

In order to evaluate the two competing models, we introduce the continuous ranked probability score (CRPS) (Matheson and Winkler, 1976):

$$\mathrm{CRPS}\left(\widehat{F}_{l|0}, y_l\right) = \int_{-\infty}^{\infty}\left(\widehat{F}_{l|0}(x) - \mathbb{I}(x < y_l)\right)^2 \mathrm{d}x, \tag{18}$$

with $\widehat{F}_{l|0}$ being the empirical cumulative distribution function (c.d.f) (Pitman, 1999) of the forecast and $\mathbb{I}(\cdot)$ being the indicator function. This metric is usually applied to probabilistic forecasts, but is equally applicable to deterministic forecasts as well (Gneiting and Raftery, 2007). In the latter case, the c.d.f becomes a step function. The overall CRPS is obtained by applying Eq. (18) to every l-step forecast and compute the average. As a second metric, we use the well-known root mean square

error (RMSE):

$$\mathrm{RMSE}\left(\widehat{y}_{l|0}, y_l\right) = \sqrt{\frac{1}{N}\sum_{l=1}^{N}\left(y_l - \widehat{y}_{l|0}\right)^2}. \tag{19}$$

When applying RMSE to a probabilistic forecast, $\widehat{y}_{l|0}$ in Eq. (19) is taken as the mean of the m ensemble members.

Summarized, we use the fitted SDE-model to generate probabilistic forecasts of both the actual stormwater response and the accumulated inflow. The former are moving l-step forecasts based on adaptive state estimation, intended for integration into an MPC framework. The latter constitute a full event simulation which is used for evaluation against the deterministic MIKE Urban model under the CRPS and RMSE metrics.

4. Results

The parameter estimates of Eq. (9) with 95%-confidence intervals (CIs) are listed in Table 2. Notably $A \approx 6.02$ km^2 is reasonable w.r.t to the total catchment area of ~ 47 km^2 because only hard surfaces like roads and roofs really contribute to the sewer inflow while green areas like parks and gardens do not. The sum of the four time constants is $\sim$3.57 h and is interpreted as the total travel time of rainfall-runoff, which is also reasonable, see Fig. 4. The 95%-CIs were first computed as symmetric Wald confidence intervals in the logarithmic domain where the parameters were estimated, and were then converted to 95%-CIs in the original parameter domain by taking the exponential. The standard errors of the parameter estimates in the logarithmic domain were derived from the diagonal elements of the inverse Hessian of the modified negative log-likelihood, under the assumption that the latter can be approximated by a second order polynomial around the optimum (Madsen and Thyregod, 2010). This assumption only holds true for parameter estimates not on the boundary, hence some of the standard errors could not be estimated and we have simply reported the associated 95%-CIs as 'NA'.

A probabilistic forecast series of the stormwater response in the test event is shown in Fig. 4 (top). It is generated by $m = 10000$ realizations of the fitted SDE-model, and the forecast horizon is 1 h, i.e. $l = 12$. Fig. 4 (center) displays a full simulation of the accumulated stormwater inflow in the same test event, based on rainfall input only. It features a comparison of the probabilistic SDE-model forecasts with the deterministic MIKE Urban forecasts. The SDE-model is here represented by the mean and 95%-prediction interval of the forecast series. The sample size is again $m = 10000$. The CRPS and RMSE of the two forecasting models are listed in Table 3. The SDE-model scores slightly better than MIKE Urban under the RMSE, and clearly better under the CRPS. The rainfall series of the test event is shown in Fig. 4 (bottom). Run time considerations and tests are included in the Supplementary material.

5. Discussion

In this paper, we have motivated, developed and estimated an SDE-based linear reservoir model with a non-linear extension. We have used the estimated model to forecast the stormwater response in the Damhus tunnel for a rainfall event unknown to the estimation process. For this test case, we have found that the model performs promisingly at 1-h-ahead forecasting, and that it is competitive with respect to the MIKE Urban model of the Damhus system, when full event simulations are compared.

The 1-h-ahead forecasts are strong in the sense that the observations generally fall within the 95%-prediction interval, while the forecast distributions are reasonably sharp at the same time. Hence, the uncertainty of the forecasts is well captured at this horizon. The reasoning behind the choice of horizon is linked to the future goal of being able to apply MPC to the system. The MPC could take advantage of the hourly changes in energy market prices, when optimizing the timing and intensity of the pumping activity. For a proof of concept, 1-h horizons are therefore considered to be relevant, but future studies may very well reveal shorter or longer horizons to be of higher importance. More details on the forecast horizons of interest are included in the Supplementary material. Every 1-h-ahead probabilistic forecast was generated in less than a second, which makes the model very feasible for non-linear MPC because a control-scheme would rely on minutely or coarser data updates.

The full event simulation setup shows that the response dynamics of the SDE-model can compete with the much more complex MIKE Urban-model. Both the SDE-model and MIKE Urban hit the response time from rainfall to stormwater in the tunnel convincingly. However, they also both overestimate the accumulated response, which could be an indication that the rainfall input series is over-representing the actual rainfall event. When compared to the performance of the moving l-step setup, this really emphasizes the strength of being able to adaptively estimate the states of the system, because over- or underpredicted rainfall responses can be corrected for. According to the RMSE and CRPS, the SDE-model outperforms MIKE Urban, at least in this particular test event. The difference is most noticeable under the CRPS, which is a consequence of the SDE-model taking the uncertainty into account. This again goes to show the potential of probabilistic forecasts over deterministic forecasts.

It is a key finding that the introduction of a simple sigmoid function to resemble the crest at the overflow structure is enough to account for the otherwise strong non-linearity, although the crest volume itself has proven somewhat tricky to estimate. The biggest liability with respect to estimation of the model in its current shape is the input. Since the catchment area is relatively large and the amount of rainfall sensors limited, the measured input is not guaranteed to be proportional to the

Table 2
Parameter estimates of the fitted SDE-model. Confidence intervals are not available for parameter estimates on the boundary of the parameter domain. Confidence intervals are available for all parameter estimates not on the boundary, except in the case of α. This is because the computed optimum is in fact a saddle point in the α-direction, and hence, its uncertainty can not be meaningfully calculated. We still consider the estimate of α to be valid, because the likelihood is sufficiently flat in the α-direction around the computed optimum to be accepted by the stopping criterion of the numerical optimization algorithm.

Parameter	Description	Estimate	95%-CI	Unit
A	Effective catchment area	6.0216	[5.4269; 6.6814]	km^2
α	Crest sharpness	6.0478	NA	–
β	Crest level (volume)	6.8968	[6.0962; 7.8027]	1000m^3
K_0	Time constant for overflow	0.0200	NA	h
K_1	Time constant for ground surface	1.5010	[1.3034; 1.7286]	h
K_2	Time constant for combined sewer	2.0440	[1.7279; 2.3903]	h
K_3	Time constant for tunnel	0.0100	NA	h
σ_1	Rainfall-related diffusion constant	0.0100	NA	1000m^3
σ_2	System-related diffusion constant	0.0924	[0.0759; 0.1123]	1000m^3
σ_e	Observation noise	0.1000	NA	1000m^3

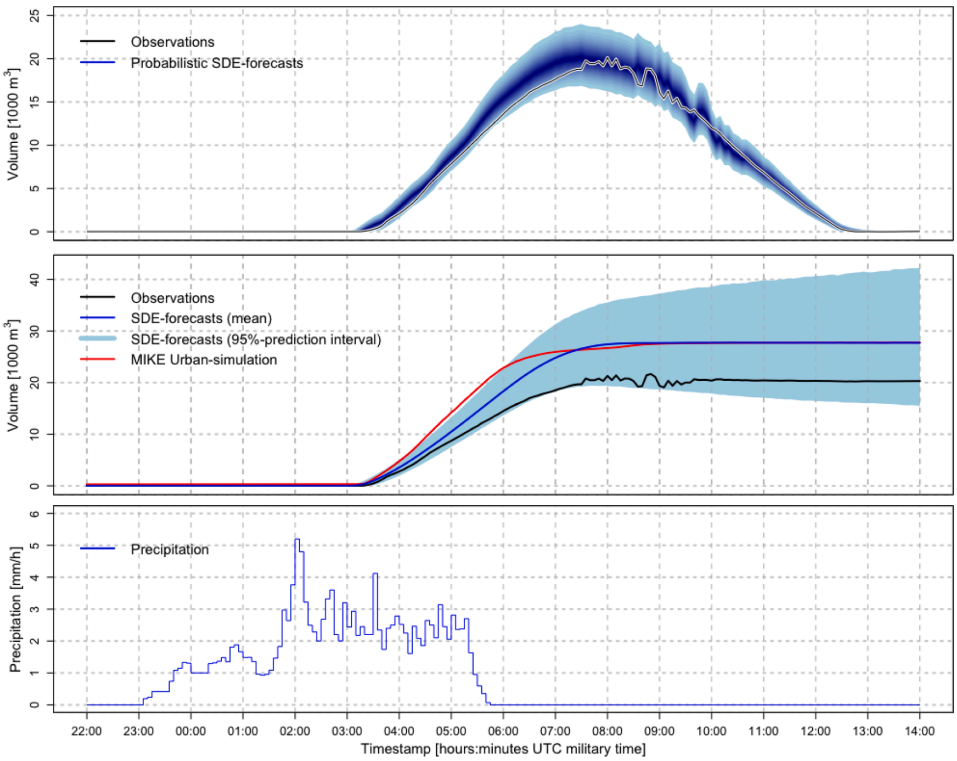

Fig. 4. a) Probabilistic forecasting by the SDE-model of stormwater in the tunnel based on rainfall input from test data. Horizon $= 1$ h. b) Comparison of the SDE-model with a deterministic MIKE Urban simulation, using accumulated inflow of stormwater as response. c) Rainfall input series. The time frame ranges from 22 October 22:00 to 23 October 14:00 in 2018 UTC military time.

Table 3
Forecast evaluation of the two models on the test data.

Method	RMSE	CRPS
SDE-based probabilistic forecasting	5.0775	2.1887
MIKE Urban deterministic forecasting	5.5021	4.3489

actual input, which in turn makes the effective catchment area and the overflow crest difficult to estimate from data consistently across different rainfall events. This problem could either be solved by increasing the number and spread of rain gauges in the catchment, or by introducing considerable noise on the input. As already discussed, however, adaptive state estimation alone can compensate for input uncertainties quite well.

Another potential improvement would be to include additional observations at other locations in the system, to increase the level of information. Especially measurements reflecting some of the states upstream of the tunnel could be useful, as in the current setup there is no information about the upstream states as long as no water has overflowed to the tunnel. This may also help estimating the crest volume more reliably. Furthermore, to keep it simple, the tunnel as modelled, is not bounded from above even though the physical tunnel does have a maximum capacity. Indeed, in the full event simulation, the upper sections in the uncertainty band of the SDE-forecasts tend to exceed the maximum capacity of 29000m^3 long-term. In a control-scheme a forecast that exceeds the physical upper bound should then translate into the same control action that a forecast equal to the physical bound would.

In this study, we have restricted ourselves to a model with a fixed set of parameters, where the success criterion is a model fit that is capable of producing reasonable probabilistic forecasts. A future study focusing on analysis of the parameters could lead to a deeper understanding of the capabilities and shortcomings of the model, and hence be grounds for improvements. Some examples of such analyses are the influence of the individual parameters, assessment of which parameters can be neglected as well as comparison with analogous parameters in other stormwater models.

Generalization of the model framework to other sewer systems should be possible. Although the specific model developed in this paper is tailored to the Damhus case, it is built from flexible and case-independent principles. Porting the model framework to a new sewer system would require an initial qualified guess on the number of reservoirs, time constants and crest functions. It would also require a basic physical understanding on how to properly embed these building bricks in the SDEs along with precipitation and pumping inputs. We believe that finding inspiration in our case-specific model would go a long way in accomplishing the above.

All in all, the Damhus case definitely demonstrates that linear reservoir models can be generalized to non-linear systems, and coupled with the SDE structure, we have provided the first stepping stone towards a smarter pumping control scheme in the Damhus system.

6. Conclusion

We have successfully developed a non-linear continuous-discrete-time state-space model for forecasting of rainfall response in a storm-water tunnel. The model is built from stochastic differential equations and is thus capable of producing probabilistic forecasts. When compared with deterministic forecasts from the MIKE Urban software, the probabilistic forecasts generated by the SDE-model are very competitive. We have also demonstrated that the SDE-model performs promisingly on 1-h horizons, thanks to the adaptive state estimation by the Extended Kalman filter. Further, since the probabilistic forecasts provide complete information about the uncertainty and can be generated in a matter of seconds, we believe this model is well-suited for integration with a future model predictive control scheme for the associated pumping system.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank HOFOR A/S for providing access to data from the Damhus system data as well as support on the technical understanding of the latter, in particular Lone B. Jørgensen. We would also like to thank Rocco Palmitessa, Morten Borup and Peter Steen Mikkelsen from DTU Environment for consulting as well as for providing access to the MIKE Urban Model used for benchmarking. Finally, we would like to thank DTU for funding.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.jhydrol.2022.127956.

References

Adams, Barry J., 2000. Urban stormwater management planning with analytical probabilistic models.
Bechmann, Henrik, Madsen, Henrik, Poulsen, Niels Kjølstad, Nielsen, Marinus K., 2000. Grey box modeling of first flush and incoming wastewater at a wastewater treatment plant. Environ.: Off. J. Int. Environ. Soc. 11 (1), 1–12.
Bjerregård, Mathias Blicher, Møller, Jan Kloppenborg, Madsen, Henrik, 2021. An introduction to multivariate probabilistic forecast evaluation. Energy AI 100058.
Borup, Morten, Grum, Morten, Madsen, Henrik, Mikkelsen, Peter Steen, 2015. A partial ensemble kalman filtering approach to enable use of range limited observations. Stoc. Environ. Res. Risk Assess. 29 (1), 119–129.
Breinholt, Anders, Thordarson, Fannar Örn, Møller, Jan Kloppenborg, Grum, Morten, Mikkelsen, Peter Steen, Madsen, Henrik, 2011. Grey-box modelling of flow in sewer systems with state-dependent diffusion. Environmetrics 22 (8), 946–961.
Brok, Niclas Laursen, Madsen, Henrik, Jørgensen, John Bagterp, 2018. Nonlinear model predictive control for stochastic differential equation systems. IFAC-PapersOnLine 51 (20), 430–435.
Cawley, Gavin C, Talbot, Nicola LC, 2003. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. Pattern Recogn. 36 (11), 2585–2592.
EFC-Sacramento, 2020. Estimating benefits and costs of stormwater management. Technical report, Environmental Finance Center at Sacramento State.
Fecarotta, Oreste, Carravetta, Armando, Morani, Maria Cristina, Padulano, Roberta, 2018. Optimal pump scheduling for urban drainage under variable flow conditions. Resources 7 (4), 73.
Gneiting, Tilmann, Katzfuss, Matthias, 2014. Probabilistic forecasting. Ann. Rev. Stat. Appl. 1, 125–151.
Gneiting, Tilmann, Raftery, Adrian E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Am. Stat. Assoc. 102 (477), 359–378.
Robert Goldstein, WEPRI Smith, 2002. Water & sustainability (volume 4): US electricity consumption for water supply & treatment-the next half century. Electric Power Research Institute.
Hansen, Lisbet Sneftrup, Borup, Morten, Møller, Arne, Mikkelsen, Peter Steen, 2014. Flow forecasting using deterministic updating of water levels in distributed hydrodynamic urban drainage models. Water 6 (8), 2195–2211.
Hsu, Wen-Ko, Tseng, Chun-Pin, Chiang, Wei-Ling, Chen, Cheng-Wu, 2012. Risk and uncertainty analysis in the planning stages of a risk decision-making process. Nat. Hazards 61 (3), 1355–1365.
Henrik Jensen, Jens Brandt Bering, 2017. Damhusledningen sætter ny standard for løsninger til spildevand.
Johansson, Rolf, Verhaegen, Michel, Chou, Chun Tung, 1999. Stochastic theory of continuous-time state-space identification. IEEE Trans. Signal Process. 47 (1), 41–51.
Rune Juhl, 2020. Statistical modelling using CTSM-R. PhD thesis.
Lund, Nadia Schou Vorndran, Falk, Anne Katrine Vinther, Borup, Morten, Madsen, Henrik, Mikkelsen, Peter Steen, 2018. Model predictive control of urban drainage systems: a review and perspective towards smart real-time water management. Crit. Rev. Environ. Sci. Technol. 48 (3), 279–339.
Madsen, Henrik, Thyregod, Poul, 2010. Introduction to general and generalized linear models. CRC Press.
Matheson, James E., Winkler, Robert L., 1976. Scoring rules for continuous probability distributions. Manage. Sci. 22 (10), 1087–1096.
Jan Kloppenborg Møller, Henrik Madsen, 2010. From state dependent diffusion to constant diffusion in stochastic differential equations by the lamperti transform.
Morari, Manfred, Lee, Jay H, 1999. Model predictive control: past, present and future. Comput. Chem. Eng. 23 (4–5), 667–682.
MOUSE Pipe Flow Reference. MOUSE Pipe Flow Reference. DHI, Hørsholm, Denmark, 2019. URLhttps://manuals.mikepoweredbydhi.help/2019/Cities/MOUSEPipeFlowReference.pdf.
MOUSE Runoff Reference Manual. MOUSE Runoff Reference Manual. DHI, Hørsholm, Denmark, 2019. URLhttps://manuals.mikepoweredbydhi.help/2019/Cities/MOUSERunoffReference.pdf.
Øksendal, Bernt, 2003. Stochastic differential equations. In: Stochastic differential equations. Springer, pp. 65–84.
Palmitessa, Rocco, Mikkelsen, Peter Steen, Law, Adrian WK, Borup, Morten, 2021. Data assimilation in hydrodynamic models for system-wide soft sensing and sensor validation for urban drainage tunnels. J. Hydroinf. 23 (3), 438–452.
Pardoux, Etienne, Talay, Denis, 1985. Discretization and simulation of stochastic differential equations. Acta Appl. Math. 3 (1), 23–47.
Pedersen, John T., Peters, John C., Helweg, Otto J., 1980. Hydrographs by single linear reservoir model. Technical report, HYDROLOGIC ENGINEERING CENTER DAVIS CA.
Pitman, Jim, 1999. Probability. Springer Science & Business Media.
Wolfgang Rauch, J.-L., Bertrand-Krajewski, Peter Krebs, Mark, Ole, Schilling, Wolfgang, Schütze, Manfred, Vanrolleghem, Peter A, 2002. Deterministic modelling of integrated urban drainage systems. Water Sci. Technol. 45 (3), 81–94.
Sabol, George V., 1988. Clark unit hydrograph and r-parameter estimation. J. Hydraul. Eng. 114 (1), 103–111.
Schmitt, Theo G., Thomas, Martin, Ettrich, Norman, 2004. Analysis and modeling of flooding in urban drainage systems. J. Hydrol. 299 (3–4), 300–311.
Schütze, Manfred, Campisano, Alberto, Colas, Hubert, Schilling, Wolfgang, Vanrolleghem, Peter A, 2002. Real-time control of urban wastewater systems-where do we stand today? In: Global Solutions for Urban Drainage, pp. 1–17.
Strelkoff, Theodor, 1970. Numerical solution of saint-venant equations. J. Hydraul. Div. 96 (1), 223–252.
Staden, Adam Jacobus Van, Zhang, Jiangfeng, Xia, Xiaohua, 2011. A model predictive control strategy for load shifting in a water pumping scheme with maximum demand charges. Appl. Energy 88 (12), 4785–4794.
Wiener, Norbert, 1923. Differential-space. J. Math. Phys. 2 (1–4), 131–174.
Zhu, Yuejian, 2005. Ensemble forecast: A new approach to uncertainty and predictability. Adv. Atmos. Sci. 22 (6), 781–788.

SUPPLEMENTARY MATERIAL

A. Information about the Damhus tunnel

A graphic showing the location of the Damhus tunnel is shown in Fig. 5 and relevant physical data are listed in Tab. IV.

Tunnel length	3.4	km
Tunnel width	3	m
Tunnel slope	-0.1044	cm/m
Invert level at Middle link	-9.26	m(DVR)
Invert level at Bottle bridge	-13.17	m(DVR)

TABLE IV

KEY SPECIFICS OF THE DAMHUS TUNNEL. M(DVR) = METERS ABOVE SEA LEVEL.

B. Information about the rainfall data

A visual representation of the optimization of the model on the 6 training events is displayed in Fig. 6. The corresponding precipitation characteristics are listed in Table V.

Event ID	Event duration (h)	Mean intensity (mm/h)	Max intensity (mm/h)	Total rainfall (mm)
1	14.9167	1.3409	5.7966	20.0019
2a	6.5833	0.8852	7.2004	5.8275
2b	0.6667	9.9000	30.000	6.6000
3	11.7500	1.1234	6.6996	13.1994
4	5.3333	1.5934	5.4000	8.4983
5	7.6666	0.7694	3.7800	5.8986
6	14.1667	0.9952	3.1392	14.0980

TABLE V

PRECIPITATION CHARACTERISTICS FOR THE 6 RAINFALL EVENTS USED FOR ESTIMATING THE SDE-MODEL. EVENT 2 IS SPLIT INTO 2 SEPARATE EVENTS IN THIS TABLE, SEE ALSO FIG. 6 (TOP-RIGHT).

C. Information about the MIKE Urban model

The MIKE Urban model is a description of every manhole, pipe, basin, outlet, crest and weir etc. in the sewer system associated with the Damhus catchment. Every such component has geographical coordinates and relevant geometry specified. Because of the relatively large size of the catchment, the number of components is large. For instance, the model contains 14261 manholes and 11320 pipes. The catchment surface area is accurately described in three dimensions.

The deterministic forecast shown in this paper is based on a two-fold simulation in MIKE Urban. First, a rainfall-runoff simulation is run in order calculate the flow of stormwater on the ground surface over the course of the rainfall event. The calculation method is based on dividing the surface area into a number of cells, and then let water discharge from cell to cell in discrete time steps, always in downstream direction. The rate at which this happens is determined by a time-area curve (Sabol, 1988). The result of the rainfall-runoff simulation contains the water volumes in each surface cell in each time step (MOUSE Runoff Reference Manual, 2019).

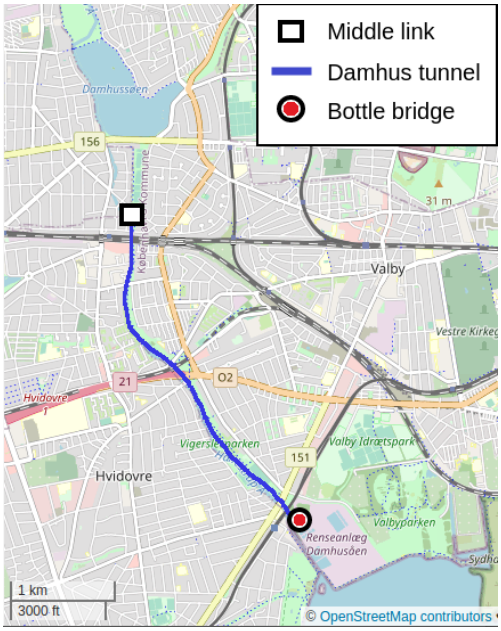

Fig. 5. Approximate location of the Damhus tunnel in Copenhagen. The image was made with OpenStreetmap (http://www.openstreetmap.org).

After completion of the rainfall-runoff simulation, the second simulation can be computed. This is called the network simulation, and it calculates the flow through the sewer network during the defined event, using the rainfall-runoff simulation result as an input (in MIKE Urban denoted a 'catchment boundary condition'). The water flow in the sewer network is modelled by the Saint Venant equations, which is a set of 2 partial differential equations describing conservation of mass and conservation of momentum (Strelkoff, 1970). The equations are solved numerically via an implicit finite difference method (MOUSE Pipe Flow Reference, 2019).

The result of the network simulation contains the water level in each network component in each time step, and thus, the results from any component can be singled out and analyzed, including the end of the stormwater tunnel, which is what we are interested in, in this study.

D. Lamperti transform

Let the $X_{i,t}$ be a random variable representing the water volume at conceptual reservoir i, and consider for instance the first of the six SDEs in Eq. (9).

$$\mathrm{d}X_{1,t} = (AU_t - \frac{2}{K_1}X_{1,t})\mathrm{d}t + \sigma_1 X_{1,t}\mathrm{d}W_{1,t} \tag{20}$$

We shall use the Lamperti transform to derive the corresponding SDE for the evolution of $Z_{1,t}$. First, we define

$$h(x,t) = \log(x), \tag{21}$$

and let

$$Z_{1,t} = h(X_{1,t}, t) = \log(X_{1,t}), \tag{22}$$

which implies that

$$X_{1,t} = h^{-1}(Z_{1,t}, t) = e^{Z_{1,t}}. \tag{23}$$

Next step is to calculate the following derivatives,

$$\frac{\partial h}{\partial t} = 0, \quad \frac{\partial h}{\partial x} = \frac{1}{x}, \quad \frac{\partial^2 h}{\partial x^2} = \frac{-1}{x^2} \tag{24}$$

Recall the we denote the drift and diffusion function of the SDE as f and g, respectively. Ito's lemma then states that

$$\mathrm{d}Z_{1,t} = \left[\frac{\partial h}{\partial x} f(x) + \frac{1}{2}\frac{\partial^2 h}{\partial x^2} g(x)^2\right]\mathrm{d}t + \frac{\partial h}{\partial x} g(x)\mathrm{d}W_{1,t}. \tag{25}$$

Inserting Eq. (24) into Eq. (25) yields

$$\mathrm{d}Z_{1,t} = \left[\frac{1}{x} f(x) - \frac{1}{2x^2} g(x)^2\right]\mathrm{d}t + \frac{1}{x} g(x)\mathrm{d}W_{1,t}. \tag{26}$$

We can then insert the explicit expressions for $f(x)$ and $g(x)$,

$$\mathrm{d}Z_{1,t} = \left[\frac{1}{x}(AU_t - \frac{2}{K_1}x) - \frac{1}{2x^2}(\sigma_1 x)^2\right]\mathrm{d}t + \frac{1}{x}\sigma_1 x \mathrm{d}W_{1,t} \tag{27}$$

Finally, we substitute $x = h^{-1}(Z_{1,t}, t) = e^{Z_{1,t}}$, reduce the result and arrive at

$$\mathrm{d}Z_{1,t} = \left[AU_t e^{-Z_{1,t}} - \frac{2}{K_1} - \frac{\sigma_1^2}{2}\right]\mathrm{d}t + \sigma \mathrm{d}W_{1,t}, \tag{28}$$

which is identical to the first SDE of the Lamperti transformed system stated in Eq. (16). The other 5 SDEs of Eq. (9) can then be Lamperti transformed using the exact same procedure. Note that the classical Lamperti transform normally implies a constant noise term in the transformed SDE, i.e. $\mathrm{d}W_t$ instead of $\sigma \mathrm{d}W_t$, so strictly speaking we are using a variant here. Yet, throughout the paper, we choose to refer to this variant as a Lamperti transform as well (Møller and Madsen, 2010).

E. Summary of the estimation algorithm

In the following, a condensed version of the model fitting algorithm used in the paper is provided. The programming language used is R. It should be seen as a summary of the key parts of the algorithm, as individual helper functions are not further documented here. The comments in the code explain what happens in each step.

```
# Load rainfall observations,
# water volumes and pumping signal
data <- loadData()

# Vector of initial parameter guesses
par_initial <- c(A,alpha,beta,K0,K1,K2,K3,
            sigma1,sigma2,simgae)

# Load parameter boundary values
par_lower <- loadLowerBounds()
par_upper <- loadUpperBounds()
```

```
# Setup SDE structure according to
# Eq. (16) and (17), details omitted here.
SDEmodel <- ctsmModelInitialize()

# Define log-likehood function
negativeloglik <- function(par_current){

  # Set parameters to current guess
  SDEmodel$par <- par_current

  # Full series of 1-step predictions
  predictions <- predict(model = SDEmodel,
                         newdata = data)

  # Calculate negative log-likelihood
  # based on the observations,
  # predicted values and
  # predicted standard deviations
  # Here "nll" is as in Eq. (13).
  result <- nll(data$y,
                predictions$y,
                predictions$sd)

  # Output the negative log-likelihood
  return(result)
}

# Optimize the negative log-likelihood
# using "nlminb" within the user-defined
# boundaries of the parameter space
fit <- nlminb(start = par_initial,
              objective = loglik,
              lower = par_lower,
              upper = par_upper)

# Extract the parameter estimates
par_fitted <- fit$par
```

F. Forecast horizons of interest

Which forecast horizons are relevant to study depends on the application at hand. In this case study, we assume that the future application will be an MPC which receives a new observation every 5 minutes. It follows, that the MPC will update its recommended future pumping inputs at that same rate (Brok et al., 2018). The shortest horizon of interest is thus 5-minute. Determining exactly how far ahead the furthest horizon of interest lies requires a deeper analysis of the Damhus system in a control setting. Yet, we at least assume that a forecast of when stormwater starts flowing into the tunnel as a consequence of precipitation, can be important information for the MPC. This horizon will depend on the characteristics of the rainfall event, so we shall make a heuristic assessment. A brief inspection of the 6 rainfall events in the training data reveals that stormwater often appears in the tunnel within 4 hours after the start of a rainfall event, see Fig. 6.

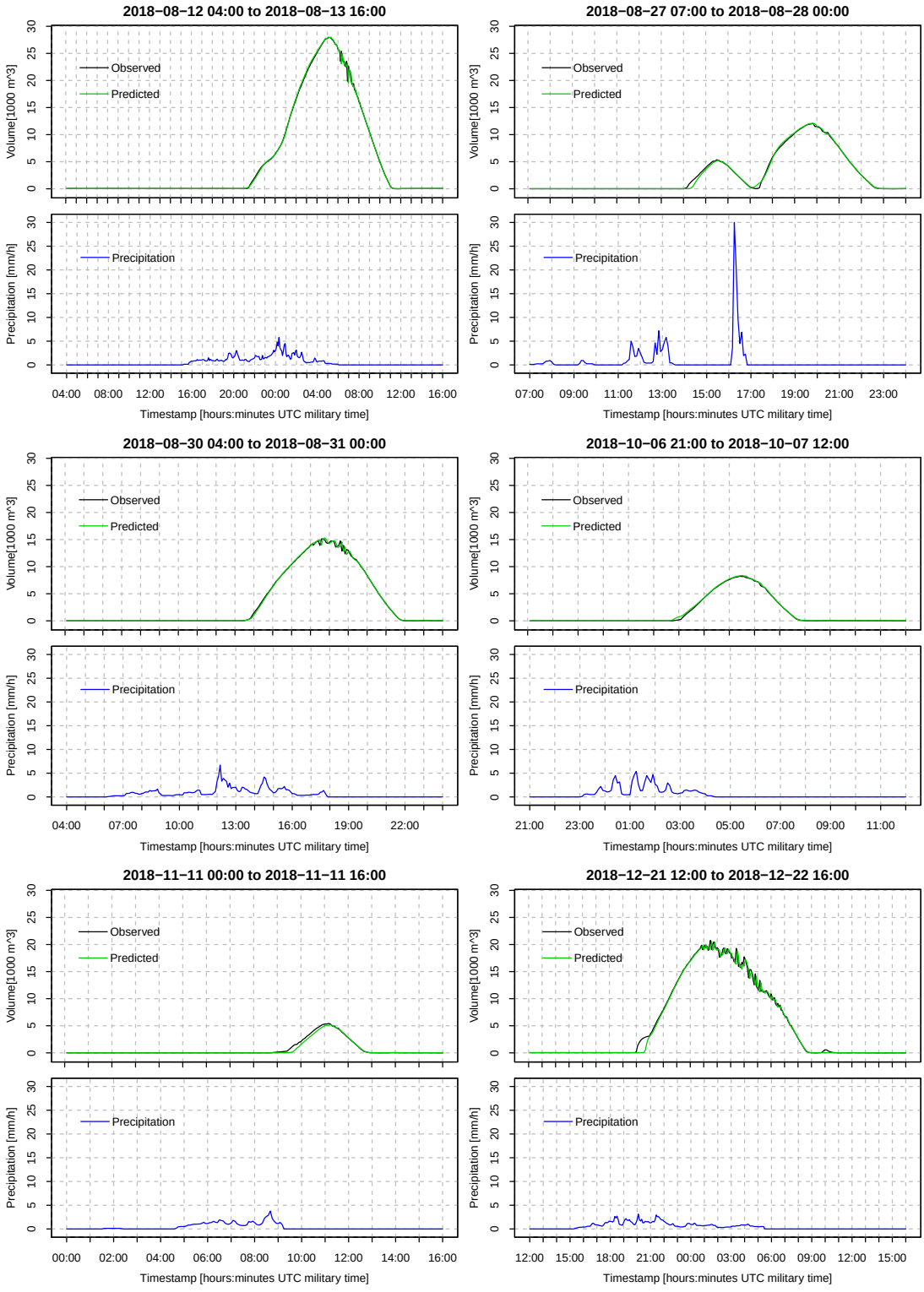

Fig. 6. Visual representation of the 6 rainfall events used for fitting the SDE-model.

We have hence decided that we are interested in horizons from 5-minute to 4-hour. For a proof of concept, we just want to show probabilistic forecasts at one horizon in between this interval. This choice can be made freely, however, the 1-hour horizon is of special interest, because the power price on the energy market changes hourly, which could easily be integrated in a future MPC application. Therefore, we choose to issue and evaluate probabilistic forecasts on 1-hour horizons.

G. Run time considerations

When testing the computational load of the forecast generation by the SDE-model, we will focus on the forecast horizon in the interval of interest that requires the longest run time, which in this case is 4-hour. Furthermore, the reported run time will be the average time needed to generate one forecast realization, i.e. $m = 1$. The expected run time scales linearly with both horizon and m, so any combination of the two can be calculated based on the reported run time. In a typical MPC setting, the SDE-model will not be evaluated as an ensemble, but will rather be integrated into an optimization scheme through its moment equations (Brok et al., 2018). Therefore, it is enough to check that $m = 1$ forecast can be generated sufficiently fast, and have this run time result be a benchmark for future studies of competing models. The test set contains 193 different 4-hour horizons. We record the run time of the forecast of every 4-hour horizon 10 times, and then repeat the whole process 30 times, producing $10 \times 30 \times 193 = 57900$ individual run time measurements. A summary of the results are reported in Table VI. Hardware specifics for the machine used for the run time tests are listed in Table VII.

We no longer have access to the MIKE Urban model nor to information about the server it was run on, when we generated its deterministic forecasts. Therefore, no controlled tests of its run time can be conducted and reported, however, from our experience, the model always took at least 20 minutes and up to more than an hour to run under these undocumented conditions. Such an order of magnitude is not feasible for a 5-minutely updated MPC.

Metric	Run time (seconds)
Mean	0.0655
Standard deviation	0.0105
Min	0.0600
Max	0.1749
Median	0.0623

TABLE VI

RUN TIME RESULTS FOR FORECASTING BY THE SDE-MODEL AT 4-HOUR HORIZON.

Operating system	Linux Mint 18.3
Kernel version	4.10.0-38-generic
Architechture	x86_64
CPU Model name	Intel(R) Core(TM) i7-7600U CPU 2.80GHz
CPU Max MHz	3900
L1d cache	32K
L1i cache	32K
L2 cache	256K
L3 cache	4096K

TABLE VII

HARDWARE SPECIFICATIONS FOR THE MACHINE ON WHICH THE RUN/RESPONSE TIMINGS OF THE SDE-MODEL FORECASTS WERE CONDUCTED.

Energy and AI 4 (2021) 100058

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Review

An introduction to multivariate probabilistic forecast evaluation

Mathias Blicher Bjerregård*, Jan Kloppenborg Møller, Henrik Madsen

Technical University of Denmark Department of Applied Mathematics and Computer, Denmark

HIGHLIGHTS

- An introduction to multivariate probabilistic forecast evaluation.
- A demonstration of how the probabilistic forecasting evaluation methods may be implemented for univariate as well as multivariate problems.
- A demonstration of how the probabilistic forecasting evaluation methods can be applied, exemplified in three case studies, with an emphasis on the evaluation on wind power forecasts.
- A summary table that highlights the advantages and drawbacks of the methods discussed.

ARTICLE INFO

Article history:
Received 20 October 2020
Received in revised form 11 February 2021
Accepted 11 February 2021
Available online 18 February 2021

Keywords:
Probabilistic forecast evaluation
Multivariate scoring rules
Wind power forecast
Ensemble forecast
Time series analysis

ABSTRACT

Probabilistic forecasting is becoming increasingly important for a wide range of applications, especially for energy systems such as forecasting wind power production. A need for proper evaluation of probabilistic forecasts follows naturally with this, because evaluation is the key to improving the forecasts. Although plenty of excellent reviews and research papers on probabilistic forecast evaluation already exist, we find that there is a need for an introduction with some practical application. In particular, many forecast scenarios in energy systems are inherently multivariate, and while univariate evaluation methods are well understood and documented, only limited and scattered work has been done on their multivariate counterparts. This paper therefore contains a review of a selected set of probabilistic forecast evaluation methods, primarily scoring rules, as well as practical sections that explain how these methods can be calculated and estimated. In three case studies featuring simple autoregressive models, stochastic differential equations and real wind power data, we implement, apply and discuss the logarithmic score, the continuous ranked probability score and the variogram score for forecasting problems of varying dimension. Finally, the advantages and disadvantages of the three scoring rules are highlighted, and this provides a significant step towards deciding on an evaluation method for a given multivariate forecast scenario including forecast scenarios relevant for energy systems.

1. Introduction

Forecast evaluation refers to the assessment of the quality of a forecast or to the selection between several competing forecasts. Traditionally, forecasters have used point forecasts [1] such as the conditional expectation for prediction of real processes. If the process is Gaussian, the uncertainty of the prediction is completely characterized by a simple symmetrical confidence interval. However, since real processes are often far from Gaussian, in order to capture all information of a process of interest, it is generally necessary to consider the entire forecast distribution. The evaluation of this is called probabilistic forecast evaluation [2].

A reliable forecast of future events is of crucial importance in, but not limited to, the design and operation of energy systems. A classic application is in the wind power sector, where the associated revenue is very dependent on reliable wind power forecasts [3]. In particular, one unexpected extreme event under which an entire wind farm is forced to shut down temporarily, can easily negate several months of revenue. This is a powerful example of why not only the expectation, but also the uncertainty of the forecasted wind power must be taken into account to minimize such a risk, ideally by forecasting the full probability distribution [4]. In order to obtain accurate probabilistic forecasts, it is necessary to have a good forecasting model, and in order to obtain the best forecasting model, it is necessary to be able to evaluate the forecasts in a meaningful way. Therefore, probabilistic forecast evaluation is clearly very important in energy systems. Besides energy systems, other examples of relevant applications include weather and climate prediction [5], economic and financial risk management [6] and epidemiological forecasting [7]. A shift from point forecasts towards probabilistic forecasts is becoming increasingly important in all of these areas [8].

Forecasting of energy systems may concern univariate or multivariate forecasts. A forecast is multivariate when it consists of multiple variables, which may refer to multiple time-steps, multiple sites or multiple parameters. Plenty of good research about probabilistic forecast evaluation has been published in the univariate case [8]. However, most practical forecast applications consider a sequence of future time points

* Corresponding author.
E-mail address: matbb@dtu.dk (M.B. Bjerregård).

https://doi.org/10.1016/j.egyai.2021.100058
2666-5468/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Nomenclature

AR(1)	Autoregressive model of order 1
ARMA(1,1)	Autoregressive-moving-average model of order 1
ARIMA(1,1,1)	Autoregressive-integrated-moving-average model of order 1
CDF	Cumulative distribution function
CdL	Conditional likelihood score
CRPS	Continuous ranked probability score
CsL	Censored likelihood score
DSS	David-Sebastiani score
LogS	Logarithmic score/log score
PDF	Probability density function
PIT	Probability integral transform
SDE	Stochastic differential equation
VarS	Variogram score

Mathematical notation

f	Probability density function
F	Cumulative distribution function
m	Number of members in an ensemble
N	Length of a time series
$\boldsymbol{X}_t$	Multivariate random variable at time t
X_t	Univariate random variable at time t
x_t	Univariate realized value at time t
$\hat{x}_t$	Univariate forecast of x_t at time t
$\{\cdot\}$	Time series of '·'

and are hence multivariate, e.g. simultaneous prediction of wind power production at different horizons. The wind power production levels at different time horizons are strongly correlated, and a proper modelling of this temporal correlation is required to obtain the best forecasts, and in turn, the highest revenue [3]. This is only possible with multivariate forecasting, so clearly, multivariate probabilistic forecast evaluation is relevant in energy systems.

In our opinion, there is a need for an introduction that allows the reader to develop an intuitive and applicable understanding of multivariate probabilistic forecast evaluation and that makes the topic more accessible to practitioners. The best way to accomplish this goal is to select the most suitable evaluation methods, and then review those methods through easy-to-follow examples with low-level calculations and illustrative content, first in the simpler, univariate setting and then eventually generalize to multivariate problems. Specifically, the methods discussed here are three performance metrics labeled the logarithmic score [9], the continuous ranked probability score [10] and the variogram score [11]. To shorten the path from theory to practical use, we also give suggestions on how these metrics can be implemented numerically. Finally, the behaviors of the evaluation methods are demonstrated and compared in three simulation studies leading to an open-ended conclusion that highlights advantages and shortcomings of each of the discussed metrics. All in all, this paper is neither a full review nor a research paper, but a hybrid with just the right balance needed to satisfy the objectives described above. Our key contribution is thus a pedagogical introduction to probabilistic forecast evaluation that builds a bridge from theory to practical implementation, with a special focus on multivariate problems.

In practice, forecasts are evaluated either qualitatively or quantitatively. The former is usually executed by applying a probability integral transform (PIT) and inspecting the resulting PIT histogram, which allows the forecaster to verify that a forecast is reasonable on its own [12]. Quantitative evaluation, on the other hand, lets the forecaster compare different competing forecasts and select the best candidate. This is done by applying *scoring rules*, i.e. functions that report a number for each forecast and hence allow direct ranking [2]. Clearly, the root mean square error is a very common example of a scoring rule. Previously, it was considered necessary to use a combination of qualitative and quantitative methods, e.g. to evaluate probabilistic wind power forecasts [13]. This involved evaluating a series of properties of the forecast one at a time, including reliability, sharpness, resolution and finally a skill score. While that framework is still used [14], it seems slightly confusing to consider multiple features of a forecast at once. For instance, how do we make a decision if different properties point towards different conclusions? With a desire to select the best forecast, it is ideal to collect all evaluation in one scoring rule instead, as long as that scoring rule is *proper*. A scoring rule applied in a forecast scenario is proper when it guarantees that the forecaster is being honest, i.e. selects the most correct forecast given the information available [2]. This property holds true for all of the scoring rules applied in this paper.

Unfortunately, no universal scoring rule is currently available that is considered to be superior for any forecasting problem, so a reasoned choice of scoring rule must be made for each individual problem. The choice of scoring rule depends on the type of variable to forecast (categorical/continuous) and the dimension (univariate/multivariate). Covering every possible scenario is far beyond the scope of this paper and hence we will narrow the focus in the following. First, we will restrict ourselves to *continuous* variables, since the majority of real forecast scenarios assume this form.

Second, we will mainly focus on evaluations methods applicable for *multivariate* forecasts, because of the importance of multivariate forecasts in energy systems as motivated above. Also, an updated review of univariate forecast evaluation already exists, while this is lacking for multivariate forecast evaluation [8]. However, we will briefly cover univariate forecast evaluation, as this is fundamental for understanding the concepts and methods.

With the focus on the evaluation of multivariate forecasts of continuous variables specified, the number of suitable evaluation methods available in the literature is limited to only a handful.

The following is an summary of published research relevant for the evaluation of multivariate probabilistic forecasts. The earliest approach concerns factorization of the forecast distribution into chains of univariate conditional distributions. Each conditional distribution is then evaluated in terms of its PIT histogram [15]. The drawback of this method, however, is that the number of PIT histograms for each forecast increases quadratically with dimension. This issue has been tackled in for the bivariate case by applying a transformation of the PIT histograms that reduces the multivariate to a univariate problem [16]. The idea of transformation has later been extended with a location-adjustment [17] and finally, a generalized PIT histogram test applicable for any dimension has been proposed [18]. Regarding multivariate scoring rules, the amount of published research is limited. Multivariate analogs of univariate scoring rules such as the logarithmic score (LogS) [9] and the continuous ranked probability score (CRPS) [10] which both evaluate the full forecast distribution have been stated in the literature, but rarely used and not deeply explored [19]. Occasionally, a generalized version of the CRPS, called the energy score has seen some application [2], but both the CRPS and the energy score have been shown to be almost useless for detecting differences in the correlation structure of the multivariate forecast [20]. To address this exact issue, the variogram score (VarS) has been suggested as an alternative [11]. Other multivariate scoring rules available in the literature include the Dawid-Sebastiani score [21], the conditional and the censored likelihood scores [22]. The most recent contribution to the list is the Schaake shuffle [23], where multivariate dependencies are evaluated by an extension of the ensemble coupla coupling method [24].

Because the focus of this paper is on probabilistic scoring rules, where the entire forecast distribution as well as the multivariate features are evaluated, we consider the LogS, the CRPS and the VarS to be most interesting methods among the options listed above and this is the reasoning behind the selection of methods for this paper.

Finally, we apply the probabilistic forecast evaluation in terms of non-parametric forecast distributions, because this approach is always

applicable and independent of the type of forecasting model. A distribution is parametric if it has a parametric representation, like the Gaussian distribution, for example, and is non-parametric otherwise. In practice, probabilistic forecasts are often issued as non-parametric distributions, e.g. when generated from ensembles [25], which in general might not be well approximated by parametric distributions, especially not in the multivariate case. On the other hand, if a probabilistic forecast does assume a parametric distribution, then it can easily be evaluated in the non-parametric framework, by sampling from it. Consequently, the non-parametric evaluation framework is not dependent on any assumptions about neither the forecast model nor the forecast distribution. The distinction between parametric and non-parametric specifically concerns the forecast distribution and should not be confused with the forecasting model itself. Even though forecast of energy systems is normally based on parametric models with physical parameters, the forecast distributions will most likely still have to be evaluated in the non-parametric framework.

The evaluation framework demonstrated in this paper comes with a few assumptions. It is assumed that the forecaster has access to a series of probabilistic forecasts or a set of forecast models from which probabilistic forecasts can be generated. It is also assumed that the forecaster is in possession of an adequate amount of data to evaluate the forecasts on. The adequate amount of data is completely dependent on the forecast problem at hand, but ideally the data-set should always reflect all the possible types of scenarios within the system which is subject to forecasting. That is the point where additional data no longer provides any new information, and hence, the conclusion of the forecast evaluation will not change significantly. In terms of computational tractability, none of the evaluation methods discussed require a certain amount of data.

The paper is organized as follows. In Section 2, we motivate the concept of scoring rules and review our selected list of those. This section also contains a very short introduction to the PIT histogram technique. In Section 3, we elaborate on how the scoring rules may be estimated numerically. The scoring rules are then applied, compared and discussed in three simulation studies in Section 4, and finally Section 5 concludes.

2. Methods

First, we introduce some notation. Let $\boldsymbol{X}_t = (X_{1,t}, \cdots, X_{k,t})^\top$ denote a k-dimensional random variable at time t. When considering a time series of length N of k-dimensional random variables, we use $\{\boldsymbol{X}_t\} = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$. When, as most often, t is implicit, we simply let $\boldsymbol{X}_t = \boldsymbol{X}$. The corresponding univariate version is $X_t = X$.

Multivariate time series may be constructed in different ways. One way is to consider the univariate time series $\{X_t\} = \{X_1, \cdots, X_N\}$ and from this construct the multivariate random variable $\boldsymbol{Z}_t = (X_{t+1}, \cdots, X_{t+k})^\top$. For example, given

$$\{X_t\} = \{X_1, X_2, \cdots, X_N\}, \tag{1}$$

we can construct the bivariate time series

$$\begin{aligned}\{\boldsymbol{Z}_t\} &= \{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_{N-2}\} \\ &= \left\{ \begin{pmatrix} X_2 \\ X_3 \end{pmatrix}, \begin{pmatrix} X_3 \\ X_4 \end{pmatrix}, \cdots, \begin{pmatrix} X_{N-1} \\ X_N \end{pmatrix} \right\},\end{aligned} \tag{2}$$

Another option is to consider two physically different univariate variables and combine them. For example, consider wind power W_t and solar power S_t. We can then construct a bivariate random variable $\boldsymbol{Z}_t = (W_t, S_t)^\top$ and subsequently get the time series

$$\{\boldsymbol{Z}_t\} = \left\{ \begin{pmatrix} W_1 \\ S_1 \end{pmatrix}, \begin{pmatrix} W_2 \\ S_2 \end{pmatrix}, \cdots, \begin{pmatrix} W_N \\ S_N \end{pmatrix} \right\}. \tag{3}$$

As seen from the two simple examples above, there are several ways to construct a multivariate time series. Throughout this article, we will exclusively consider the former case, such that 'multivariate' always means multivariate w.r.t. time.

Since we are dealing with probabilistic forecasts, we frequently encounter the cumulative distribution function (CDF) of $\boldsymbol{X}$

$$F_{\boldsymbol{X}}(x_1, \cdots, x_k) = P(X_1 \leq x_1, \cdots, X_k \leq x_k), \tag{4}$$

as well as the probability density function (PDF)

$$f_{\boldsymbol{X}}(x_1, \cdots, x_k) = \frac{\partial^k}{\partial x_1 \cdots \partial x_k} F_{\boldsymbol{X}}(x_1, \cdots, x_k). \tag{5}$$

Often, these functions are simply referred to as F and f, respectively. Finally, we will occasionally need the indicator function $\mathbb{I}(\omega)$, i.e. $\mathbb{I}(\omega) = 1$ if the statement ω is true and 0 otherwise.

Forecasts may be evaluated either qualitatively or quantitatively. For qualitative evaluation, we issue the probability integral transform (PIT, cf. Section 2.2); for quantitative evaluation, we shall introduce the concept of scoring rules.

2.1. Scoring rules

A scoring rule $S(G, \boldsymbol{y})$ is a function of any forecast G (e.g. a point forecast, a quantile forecast or a PDF) and an observation $\boldsymbol{y}$. The scoring evaluates to a summary measure, which we denote the *score*. Given a time series $\{\boldsymbol{y}_t\} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N\}$, every pair of forecast and corresponding realized observation $(G_i, \boldsymbol{y}_i)$ is evaluated, and the overall score of the model is then usually reported as the average score, i.e.

$$\bar{S}(G, \boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} S(G_i, \boldsymbol{y}_i) \tag{6}$$

although alternative weights like an exponential decay [26]

$$S_N(G, \boldsymbol{y}) = (1 - \lambda) S(G_i, \boldsymbol{y}_i) + \lambda S_{N-1}(G, \boldsymbol{y}) \quad , \quad 0 < \lambda < 1, \tag{7}$$

may be used if desired. Scoring rules can be regarded as *loss* functions, such that smaller scores are preferred. Hence, a meaningful scoring rule should be constructed such that we can build an optimizer around it.

2.1.1. Logarithmic score

Let us first consider a univariate time series $\{y_t\}$ and the evaluation of a univariate density forecast, i.e. $G = f$. From likelihood theory, the *logarithmic score*, $S(G, y) = \text{LogS}(f, y)$ naturally emerges [9]. It is based on the PDF and defined as

$$\text{LogS}(f, y) = -\log f(y). \tag{8}$$

Elaboration on the practical use of LogS is found in Section 3.1. Clearly, LogS rewards models under which the observed event is likely to occur, i.e. has high probability. It is optimal at the mode of f, as exemplified in Fig. 1. When considering the multivariate forecast $f_{\boldsymbol{X}}$ and the observation $\boldsymbol{y}$, the multivariate analog of LogS is simply

$$\text{LogS}(f_{\boldsymbol{X}}, \boldsymbol{y}) = -\log f_{\boldsymbol{X}}(\boldsymbol{y}). \tag{9}$$

LogS is thus equivalent to the log-likelihood of the forecast model, and it enjoys ideal properties, as it captures all possible information about the observations in relation to the model, including the correlation between multivariate forecasts. However, this scoring rule has a potentially significant drawback, namely that it penalizes unlikely observations very hard. Therefore, tiny changes to the tails of a density forecast can result in a dramatic change in LogS, even when the overall shape of the density is unchanged. This behavior is illustrated in case study No. 3, cf. Section 4.3 and may be stabilized by tuning down the emphasis on unlikely observations, cf. Section 2.1.4.

Example 1: calculation of LogS

In the following, we apply scoring rules to two models and a simple data set, in order to illustrate how scoring rules work and what they reward.

Consider a probabilistic forecast given by the Beta distribution, which can be parametrized as follows:

$$f(y) = \frac{1}{\text{B}(\alpha, \beta)} y^{\alpha - 1} (1 - y)^{\beta - 1}. \tag{10}$$

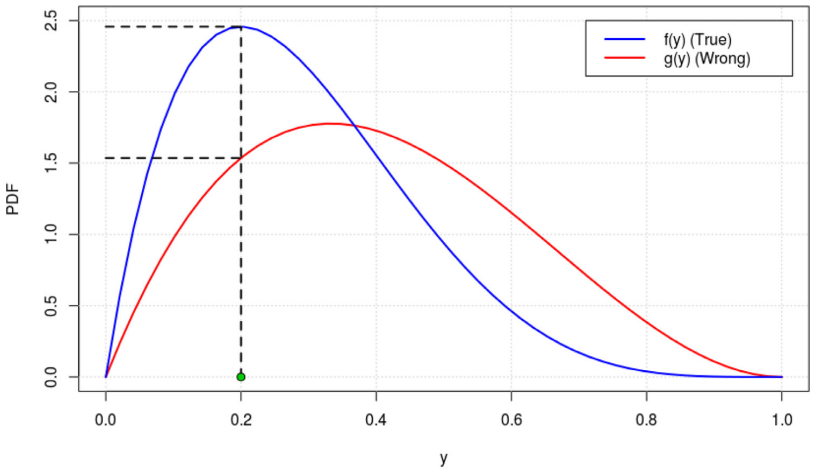

Fig. 1. Probability density at the mode of Beta(2,5) ($y = 0.2$, green dot) under the true model (blue) and a wrong model (red), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

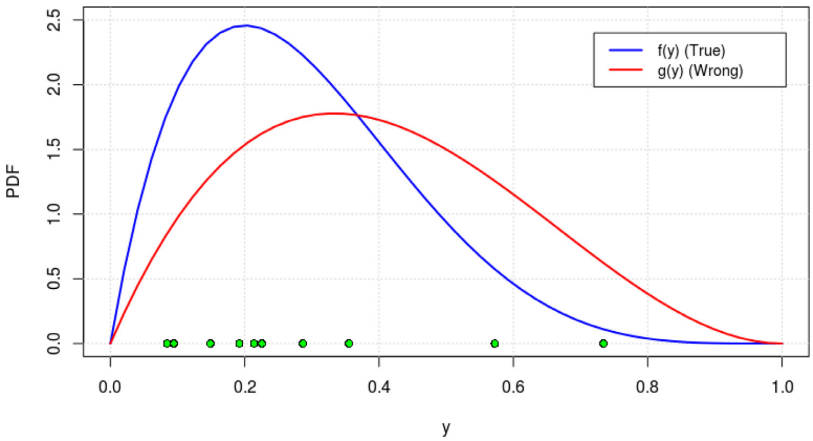

Fig. 2. 10 observations (green dots), simulated from a Beta(2,5)-distribution. The true model (blue) is shown along with the PDF for a Beta(2,3)-distribution (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We set $(\alpha, \beta) = (2, 5)$ and simulate 10 observations, $y_1, y_2, ..., y_{10}$ which are shown in Table 1. Hence, f is the true model. Let $g = \text{Beta}(2, 3)$ be a competing - and obviously wrong – model. Both models as well as the observations are shown in Fig. 2.

LogS of model f for the first observation, $y_1 = 0.149$ is calculated,

$$\begin{aligned} -\log f(y_1) &= -\log f(0.149) \\ &= -\log[\text{B}(2,5)^{-1} 0.149^{2-1}(1-0.149)^{5-1}] \\ &= -0.852. \end{aligned} \tag{11}$$

All the other individual LogS are calculated in the same way and presented in Table 2. 8 out of 10 of the observations are more likely to occur under the true model, and the latter will also be favored when we calculate the final key quantity: the average LogS, cf. Eq. (7),

$$\begin{aligned} \overline{\text{LogS}}(f, \boldsymbol{y}) &= -0.34, \\ \overline{\text{LogS}}(g, \boldsymbol{y}) &= -0.23. \end{aligned} \tag{12}$$

Hence, as expected f provides the best density forecast.

2.1.2. Continuous ranked probability score

Now, consider again the univariate time series $\{y\}$, but let the probabilistic forecast take the form of a CDF, i.e. $G = F_Y$. We can then apply the *continuous ranked probability score* [10], $S(G, y) = \text{CRPS}(F, y)$,

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(u) - \mathbb{I}(u \geq y))^2 \text{d}u. \tag{13}$$

Elaboration on the practical use of CRPS is found in Section 3.2. CRPS is based on the forecast CDF, F and measures the squared distance between the observation and the median of F. Hence, the model is rewarded for observations close to its median.

To get a feeling of what is going on, consider the true F to be Beta(2,5) and a wrong model to be Beta(1,5). The true model has its median equal to 0.264. Hence, given the observation $y = 0.264$, the CRPS of this observation would be minimized under the true model, with the wrong model being inferior. Let us split the integral in Eq. (13) into two

Table 1
10 simulated observations following a Beta(2,5)-distribution.

i	1	2	3	4	5	6	7	8	9	10
y_i	0.149	0.095	0.287	0.355	0.226	0.192	0.214	0.734	0.572	0.084

Table 2
Log scores of the two different beta-models w.r.t. the 10 individual observations. The superior scores for individual observations are highlighted with bold.

i	1	2	3	4	5	6	7	8	9	10
LogS(f, y_i)	**−0.85**	**−0.65**	**−0.80**	**−0.61**	**−0.89**	**−0.90**	**−0.90**	2.21	0.55	**−0.58**
LogS(g, y_i)	0.26	0.07	0.56	0.57	0.48	0.41	0.46	**0.47**	**−0.23**	0.16

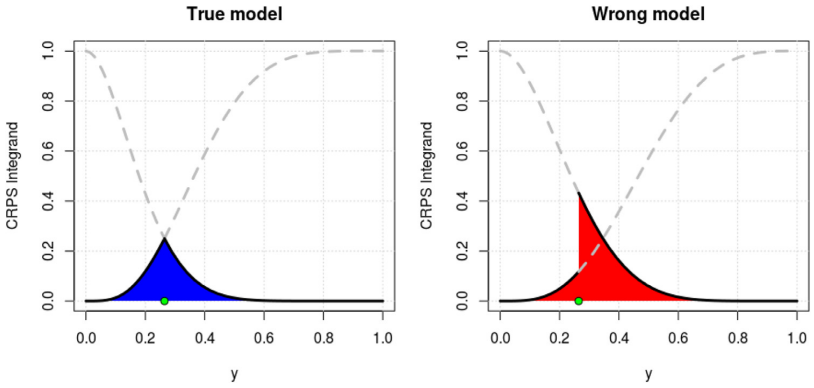

Fig. 3. CRPS of the median of Beta(2,5) ($y = 0.264$, green dot) under the true model (left) and the wrong model (right), respectively. The solid black line is the CRPS integrand and the CRPS is equal to the colored area under the curve. The two parts of the integrand are illustrated fully as if they were evaluated in the entire range from 0 to 1 (non-evaluated part shown as grey dotted line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

parts,

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{y} F(u)^2 \mathrm{d}u + \int_{y}^{\infty} (F(u) - 1)^2 \mathrm{d}u. \tag{14}$$

Then it is seen that the two integrands intersect at the median of F, which is illustrated in Fig. 3. Thus, it is easy to see why observations close to the median are rewarded, as this minimizes the area under the curve, i.e. minimizes CRPS.

The multivariate generalization of CRPS is given by

$$\mathrm{CRPS}(F, \boldsymbol{y}) = \int_{-\infty}^{\infty} (F(\boldsymbol{u}) - \mathbb{I}(\boldsymbol{u} \geq \boldsymbol{y}))^2 \mathrm{d}\boldsymbol{u}. \tag{15}$$

As LogS, CRPS has both advantages and drawbacks. In terms of its abilities as a scoring rule, CRPS is stable with respect to similar models, i.e. seemingly similar models are determined to be similar, while LogS penalizes small differences in the probability tails very hard, even though these differences may not be of importance to the forecaster. As a drawback, CRPS does a poor job of detecting misspecified correlation [20], and this will be dealt with in Section 2.1.3. Regarding numerical computation, CRPS has an advantage in being based on the CDF, because the latter is faster and more robust to estimate than the PDF. Therefore, it is quite fast to compute for lower dimensions, but since a part of the computation involves the estimation of an integral, efficient computation becomes increasingly challenging with higher dimensions. The choice of numerical integral approximation method thus has a huge impact on computation in higher dimensions.

Example 2: calculation of CRPS

Recall the simple calculation example from Section 2.1.1 featuring a Beta(2,5)-distributed random variable. We will now calculate the CRPS for this example. For this purpose, we need the CDF of the beta distribution,

$$F(y) = \frac{1}{\mathrm{B}(\alpha, \beta)} \int_0^y u^{\alpha-1}(1-u)^{\beta-1} \mathrm{d}u. \tag{16}$$

Since Beta(α, β) has support [0,1], the CRPS readily reduces to

$$\mathrm{CRPS}(F, y) = \int_0^1 (F(u) - \mathbb{I}(u \geq y))^2 \mathrm{d}u. \tag{17}$$

Which may be split into two integrals seperated at x,

$$\mathrm{CRPS}(F, y) = \int_0^y F(u)^2 \mathrm{d}u + \int_y^1 (F(u) - 1)^2 \mathrm{d}u. \tag{18}$$

For simplicity, we do not attempt to reduce further but simply evaluate (18) numerically. Subsequently, all the CRPS for the individual data points are presented in Table 3. Again, the majority of the observations are most likely to occur under the true model, although the difference appears to be less significant compared to the conclusion of the log score. The average CRPS are calculated to

$$\begin{aligned} \overline{\mathrm{CRPS}}(f, \boldsymbol{y}) &= 0.11, \\ \overline{\mathrm{CRPS}}(g, \boldsymbol{y}) &= 0.13. \end{aligned} \tag{19}$$

Even though CRPS agrees with LogS that the true model is superior to the wrong model, CRPS evaluates the two models to be more similar than LogS does. Note that this is only a small example with just 10 observations. With larger sample sizes, differences between competing models according to CRPS will be more significant.

2.1.3. Variogram score of order p

Consider the multivariate time series $\{\boldsymbol{y}_t\}$ and the evaluation of a multivariate density forecast, $G = f_{\boldsymbol{X}}$. In order to address the problem of proper detection of correlation structure, we introduce the *variogram score* of order p [11]:

$$\mathrm{VarS}_p(f_{\boldsymbol{X}}, \boldsymbol{y}) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} w_{ij} (|y_i - y_j|^p - E[|X_i - X_j|^p])^2. \tag{20}$$

Elaboration on the practical use of VarS is found in Section 3.3. VarS_p is based on pairwise differences of the components of the multivariate forecast. For example, if we consider the arbitrary three-dimensional forecast

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}, \tag{21}$$

then we apply VarS over the three unique pairs, (X_1, X_2), (X_1, X_3) and (X_2, X_3). When we consider a forecast that is multivariate in terms of time, the summation indices i and j then refer to forecast horizons. The parameter p can be tuned to transform the distribution of absolute differences into being closer to symmetric than when untransformed, which enhances model separation efficiency as well as the sampling properties of $|X_1 - X_2|^p$. Setting $p = 0.5$ is a good choice for this matter [11], at least for Gaussian distributions, cf. Fig. 4. The impact of the individual

Table 3
CRPS of the two different beta-models, f and g w.r.t. the 10 individual observations. The superior scores for individual observations are highlighted with bold.

i	1	2	3	4	5	6	7	8	9	10
CRPS(f, y_i)	**0.07**	**0.11**	**0.04**	0.06	**0.04**	**0.05**	**0.05**	0.36	0.21	**0.12**
CRPS(g, y_i)	0.15	0.19	0.07	**0.05**	0.10	0.12	0.10	**0.23**	**0.11**	0.20

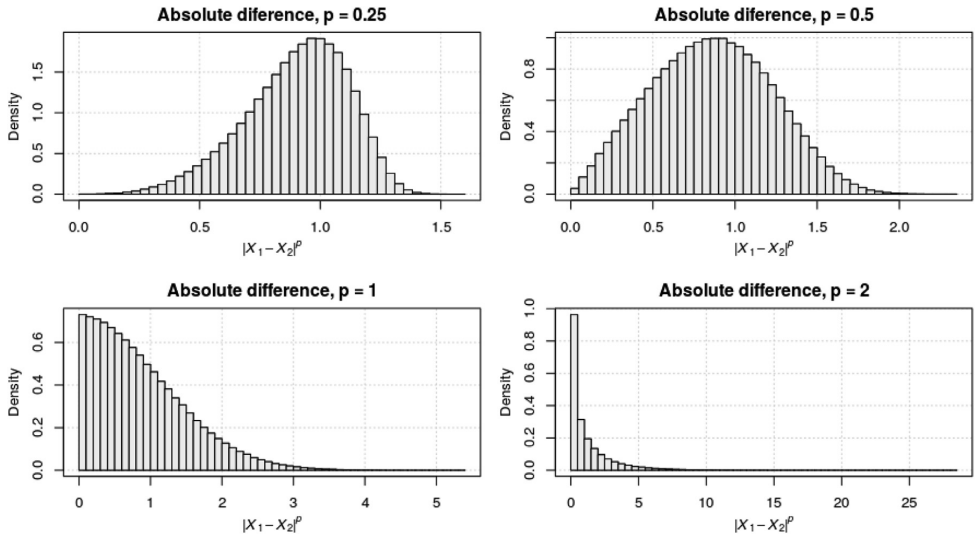

Fig. 4. Distribution of absolute differences to the power p, $|X_1 - X_2|^p$, where $(X_1, X_2)^\top$ follows a bivariate Gaussian distribution.

pairs may be adjusted by tuning the weights, w_{ij}. As a generic choice, given any pair of components, it is reasonable to let the corresponding weight be proportional to the inverse distance between the components [11]. However, throughout this paper we will use identity weights for simplicity.

Example 3: a closer look at the properties of VarS

We will investigate the bivariate Variogram Score of order $p = 0.5$ in four different cases. As seen from Eq. (20), in terms of the forecast model, VarS depends solely on a function of the expected absolute difference. Therefore VarS is optimized along the two straight lines $y_2 = y_1 \pm E|X_2 - X_1|$ (cf. Fig. 5a) and the line $y_2 = y_1$ consists of local maxima, except in the trivial case where the model is $f_X = \mathbf{0}$ which never occurs in real applications.

Bivariate normal distribution with different means and positive correlation

Let

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} 7 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 & 4 \\ 4 & 4 \end{pmatrix}\right]. \tag{22}$$

X_1 and X_2 have an expected absolute difference ≈ 6.01. Fig. 5b shows a contour plot of VarS for the forecast distribution of $(X_1, X_2)^\top$ along with sample observations generated from that same distribution. It is seen that if this distribution has its mean shifted by $(\alpha, \alpha)^\top$ with $\alpha \in \mathbb{R}$, VarS is unchanged for any observation. Furthermore any of these optimal distributions may be mirrored in the line $y_2 = y_1$ and fully preserve their respective VarS. The main point is, the true forecast distribution is only one among infintely many with an optimal VarS.

Bivariate normal distribution with equal means and positive correlation

Let

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 & 4 \\ 4 & 4 \end{pmatrix}\right]. \tag{23}$$

The expected absolute difference is ≈ 1.13. This example is similar to the first one, with the main difference that the means of X_1 and X_2 are equal. This has the funny consequence that the line $y_2 = y_1$ "intersects" with the distribution, as do both lines of optimality, c.f. Fig. 5c. Hence, many observations close to the center of the distribution, i.e. observations that are very likely to occur, are to some extent penalized. This would not happen under e.g. the logarithmic score. Generally, this penalizing behavior is expected to occur whenever the line $y_2 = y_1$ intersects with the forecast distribution, which happens when the means of X_1 and X_2 are sufficiently close to be equal.

Bivariate normal distribution with equal means and negative correlation

We repeat the previous example, except we "flip" the covariance matrix, to get negative correlation, i.e.

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 & -4 \\ -4 & 4 \end{pmatrix}\right]. \tag{24}$$

The expected absolute difference is ≈ 3.41. The resulting VarS plot is seen in Fig. 5d. Here, VarS evaluates the forecast distribution in a way that may appear odd to the observer. Most observations fall within regions of high reward (dark regions in 5 d), however, a small area around the center of the distribution is penalized, as are the outer tails. From a quick glance one could easily get the impression that a distribution carefully "placed" within the regions of high reward would yield a higher VarS than the true forecast distribution. The fact that VarS is a proper scoring rule, however, will prevent this, as is illustrated in the next example.

Sensitivity of VarS to variance and correlation

The final example demonstrates that, given that the mean and variance are already correctly calibrated, VarS is minimized when the correlation structure is correct. At the same time, we provide an example on

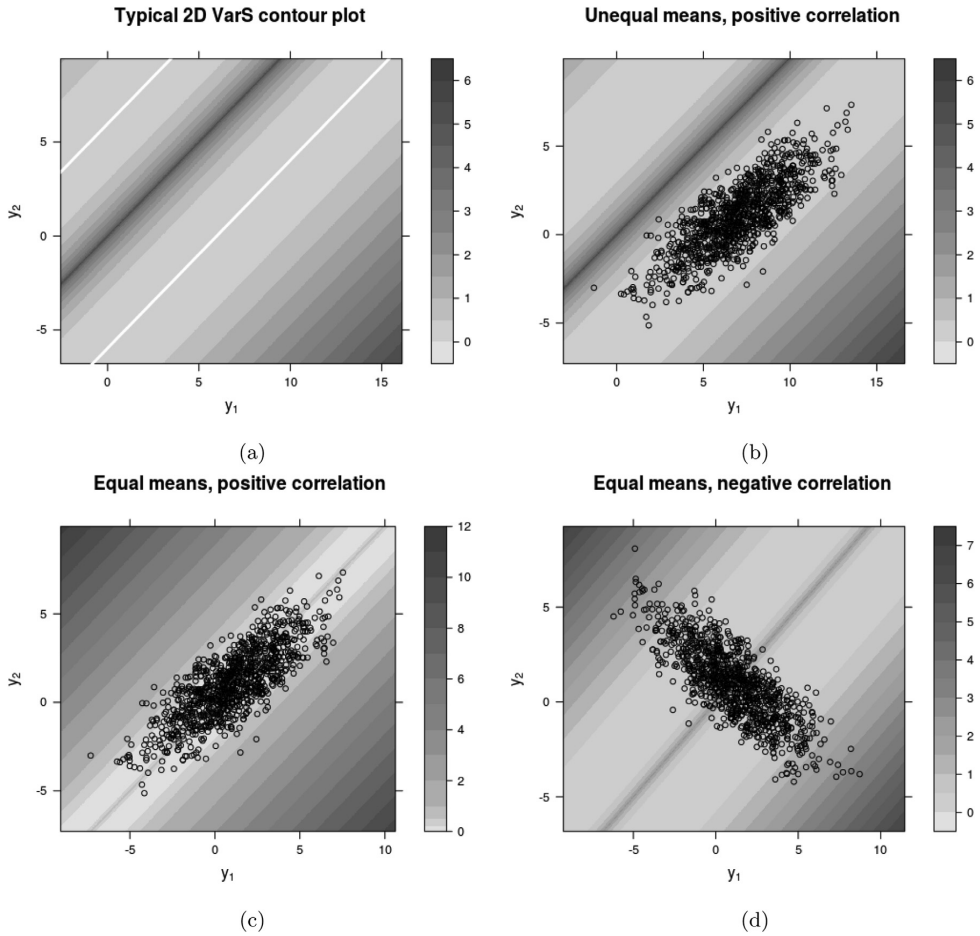

Fig. 5. Contourplot of VarS of order 0.5 for bivariate normal forecast densities. Sample observations are shown as black circles. (a): the two lines of optimality highlighted in white; (b): unequal means $\mu = (7, 1)^\top$, positive correlation; (c): equal means ($\mu = (1, 1)^\top$), positive correlation; (d): equal means ($\mu = (1, 1)^\top$), negative correlation, identical to (c) apart from the sign of the correlation.

how VarS can be calculated semi-analytically, which does at present not appear to be published in the literature. Consider the random variable, $X = X_1 - X_2$, where

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}\right], \tag{25}$$

then

$$X \sim N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2 - 2\sigma_{12}). \tag{26}$$

In the special case where $\mu_1 = \mu_2$, it can be shown (see proof in the Appendix) that the expectation of $|X_1 - X_2|^p$ with $p = 0.5$ is:

$$\mathrm{E}[|X_1 - X_2|^p] = 4 \int_0^\infty \frac{u^2}{\sqrt{2\pi\sigma_X^2}} \mathrm{e}^{-\frac{(u^2-\mu_X)^2}{2\sigma_X^2}} \mathrm{d}u. \tag{27}$$

with $\mu_X = \mu_1 - \mu_2$ and $\sigma_X^2 = \sigma_1^2 + \sigma_2^2 - 2\sigma_{12}$. Now, let ρ be the correlation of $(X_1, X_2)^\top$ and let $\mu_1 = \mu_2 = 0, \sigma_1^2 = \sigma_2^2 = 2$ (we simply write $\sigma^2 = 2$) and $\rho = 0.7$. Then VarS is calculated for all $\sigma^2 \in (0, 3)$ and $\rho \in (-1, 1)$ by substituting Eq. (27) into Eq. (20). The resulting contour plot is showed in Fig. 6a. Here it is verified that if σ^2 is correct from the beginning, then the true ρ will indeed minimize VarS as expected. However, if σ^2 is wrongly estimated, then ρ will also turn out wrong. The analogous result for the same case with $\rho = -0.7$ is shown in Fig. 6b and yields the same conclusion.

Summary of VarS properties

When using the variogram score, one should keep in mind that the true forecast distribution is not unique in terms of optimality. In fact, there are (under VarS) infinitely many optimal forecast distributions for a given forecast scenario. Also, forecast distributions that intersect with the identity line are subject to a penalty that would not occur under traditional scoring rules like LogS or CRPS. Furthermore, negatively correlated forecast distributions are evaluated in a strange way, where many very probable observations are likewise penalized. All of the above, however, are not problematic as long as the forecast density is already calibrated w.r.t. mean and variance.

2.1.4. Other multivariate scoring rules

In the following, we list a brief summary of other multivariate scoring rules proposed in the literature. The *Dawid-Sebastiani score* [21] only depends on the first two moments of the predictive distribution, i.e. the

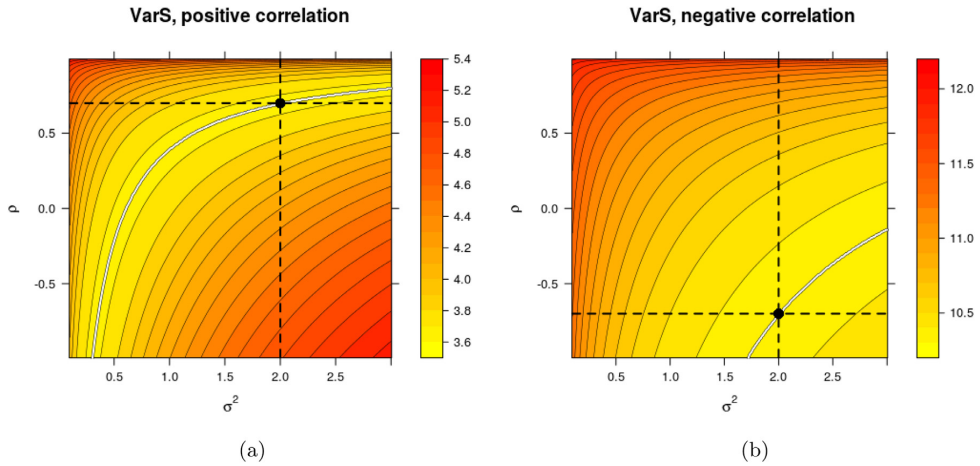

Fig. 6. Contourplot of VarS of order 0.5 for a forecast of a bivariate normal variable on the form Eq. (24) with $\mu_1 = \mu_2 = 0$, $\sigma_1^2 = \sigma_2^2 = 2$ and (a): $\rho = 0.7$; (b): $\rho = -0.7$. Minimum given σ^2 is showed as a light blue curve. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$,

$$\mathrm{DSS}(f, \boldsymbol{y}) = -\log\det(\boldsymbol{\Sigma}) - (\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}). \tag{28}$$

If the forecast is Gaussian, then, apart from an additive constant, DSS is proportional to LogS. However, several examples where DSS fails to pick the correct model have been published [11]. If one only wishes to evaluate a certain region A of the predictive distribution, the *conditional likelihood score* [22],

$$\mathrm{CdL}(f, \boldsymbol{y}) = -\mathbb{I}(\boldsymbol{y} \in A)\log\left(\frac{f(\boldsymbol{y})}{\int_A f(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}\right), \tag{29}$$

is suggested, which is essentially an extension of LogS where observations outside the desired probability region are simply ignored. For instance, it may be used to counteract the hard penalization of small probabilities (cf. Section 2.1.1) while preserving the advantages of LogS. To achieve such a compromise without completely ignoring the occurrence of observations outside of A, the *censored likelihood score* [22],

$$\mathrm{CsL}(f, \boldsymbol{y}) = -\mathbb{I}(\boldsymbol{y} \in A)\log f(\boldsymbol{y}) - \mathbb{I}(\boldsymbol{y} \in A^c)\log\left(\int_{A^c} f(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right), \tag{30}$$

may be used instead, where A^c is the complement of A.

2.2. Probability integral transform-based evaluation

While scoring rules perform quantitative evaluation, forecasting models can also be evaluated qualitatively by inspecting *probability integral transform* (PIT) histograms. Starting with the univariate case, given an observation y_t at time t and a forecast density, f, the PIT, z_t, is defined as

$$z_t = \int_{-\infty}^{y_t} f(u)\mathrm{d}u. \tag{31}$$

If the f is correctly calibrated, then $Z_t \sim$ i.i.d $U(0,1)$ [12]. Hence, the model can be validated by applying this transformation for all observations, then constructing and inspecting the PIT histogram and verifying qualitatively that the z_t series does not invalidate the uniformity assumption. A few examples of typical PIT histograms are illustrated in Fig. 7.

For multivariate forecast evaluation, the density is split into independent, conditional densities, which can then be checked individually for uniformity. For example, for two dimensions, we can factor the density of $\boldsymbol{y}_t = (y_{1,t}, y_{2,t})^\top$ in two ways,

$$\begin{aligned} f(\boldsymbol{y}_t) &= f(y_{2,t}|y_{1,t})f(y_{1,t}), \\ f(\boldsymbol{y}_t) &= f(y_{1,t}|y_{2,t})f(y_{2,t}) \end{aligned} \tag{32}$$

Each of the four densities, $f(y_{1,t})$, $f(y_{2,t})$, $f(y_{2,t}|y_{1,t})$ and $f(y_{1,t}|y_{2,t})$ can then be transformed into its respective PIT series by Eq. (31). For a reasonable model, all four of these PIT series should then be i.i.d $U(0,1)$ [15]. Theoretically, this approach can be extended to any dimension, but in practice it becomes less practical with increasing dimension. A more advanced PIT-based test that is practically applicable for arbitrary dimension is available in the literature [18].

3. Applied non-parametric forecast evaluation

In the following, we give elaborate suggestions for how the scoring rules of concern may be applied to non-parametric probabilistic forecasts. The corresponding R code is available on Github.[1] Throughout the entire section, we have the following setup. We consider the univariate time series of N observations

$$\{y_t\} = \{y_1, y_2, \cdots, y_N\}, \tag{33}$$

and we want to forecast up to k horizons ahead, i.e. a forecast of

$$\boldsymbol{y}_t = (y_{t+1}, y_{t+2}, \cdots, y_{t+k})^\top. \tag{34}$$

For that purpose we construct the multivariate series

$$\begin{aligned} \{\boldsymbol{y}_t\} &= \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_{N-k}\} \\ &= \left\{\begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_{1+k} \end{pmatrix}, \begin{pmatrix} y_3 \\ y_4 \\ \vdots \\ y_{2+k} \end{pmatrix}, \cdots, \begin{pmatrix} y_{N-k+1} \\ y_{N-k+2} \\ \vdots \\ y_N \end{pmatrix}\right\}. \end{aligned} \tag{35}$$

Consider first

$$\boldsymbol{x}_i = \hat{\boldsymbol{y}}_t^{(i)} = (\hat{y}_{t+1|t}^{(i)}, \hat{y}_{t+2|t}^{(i)}, \cdots, \hat{y}_{t+k|t}^{(i)})^\top, \tag{36}$$

[1] https://github.com/matbbDTU/probforecasteval.

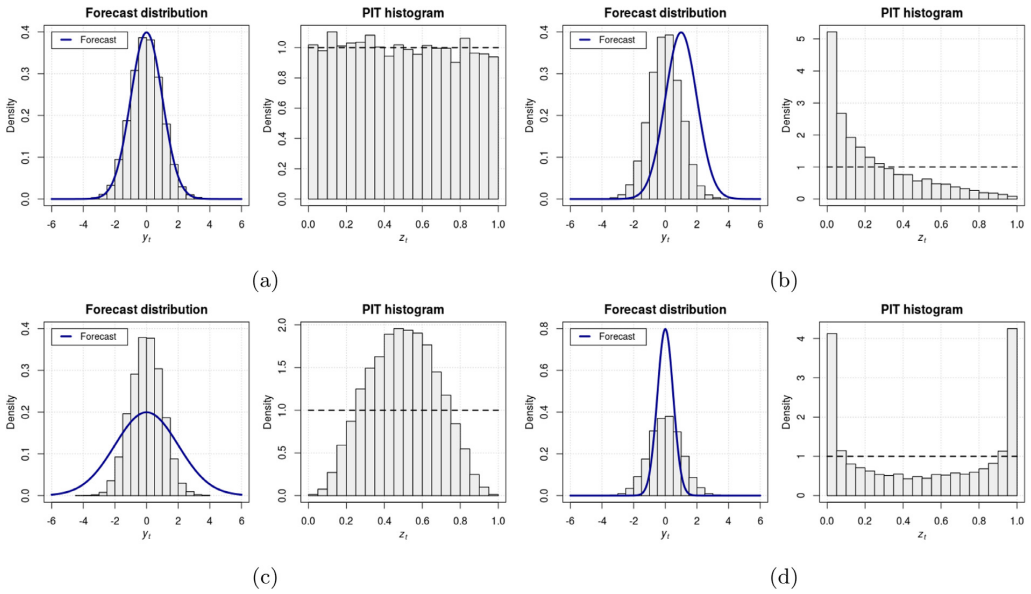

Fig. 7. PIT histograms of a fixed times series $\{y_t\}$ under the true model (a), a mean-shifted, hence miscalibrated model (b), an underdispersed model (c) and an overdispersed model (d). Only the true model with the correct mean and shape of the PDF yields a PIT histogram that appears to be uniform.

i.e. the i'th point forecast of $\boldsymbol{y}_t$. We can then collect m point forecasts to obtain an *ensemble* $\boldsymbol{x}$ of m forecast members,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_m^\top \end{pmatrix} = \begin{pmatrix} \hat{y}_{t+1|t}^{(1)} & \hat{y}_{t+2|t}^{(1)} & \cdots & \hat{y}_{t+k|t}^{(1)} \\ \hat{y}_{t+1|t}^{(2)} & \hat{y}_{t+2|t}^{(2)} & \cdots & \hat{y}_{t+k|t}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{t+1|t}^{(m)} & \hat{y}_{t+2|t}^{(m)} & \cdots & \hat{y}_{t+k|t}^{(m)} \end{pmatrix}, \tag{37}$$

which forms a numerical representation of a probabilistic forecast. Suppose, we want to evaluate a series of probabilistic forecasts in terms of its marginal distributions with the LogS in R. If `obs` is a vector of `N` observations, and `x` is an `m` x `N` matrix of `m` ensemble forecast members each of length `N`, then a simple forecast evaluation framework could be implemented as shown below:

```
# Forecast evaluation of the data with the log-score.
for(t in 1:N){

  f <- x[,t]  # Marginal density at time t
  y <- obs[t] # Observation at time t
  scores[t] <- logs(f,y)

}

# Report the average score for the model and the data-set.
mean(scores)
```

More detail on how the individual scores can be implemented is supplied in the following subsections.

3.1. Logarithmic score

Since LogS is based on the conditional density, this density has to be estimated. This can be done by using a *kernel density estimate* [27]

$$\hat{f}_h(y) = \frac{1}{mh} \sum_{i=1}^{m} K\left(\frac{y - x_i}{h}\right), \tag{38}$$

where $x_i = y^{(i)}$, h is the bandwith and K is a chosen kernel function, i.e. a symmetric function with

$$\int_{-\infty}^{\infty} K(u)du = 1. \tag{39}$$

Many possible choices of kernels exist, e.g. the Gaussian kernel

$$K(u) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}u^2} \quad , \quad u \in \mathbb{R}, \tag{40}$$

or the commonly chosen Epanechnikov kernel [28]

$$K(u) = \frac{3}{4}(1 - u^2) \quad , \quad |u| < 1. \tag{41}$$

By applying this method, we are able to obtain a non-parametric approximation of the conditional density.

The multivariate analog of Eq. (38) is given by

$$\hat{f}_{\boldsymbol{H}}(\boldsymbol{y}) = \frac{1}{m\sqrt{\det(\boldsymbol{H})}} \sum_{i=1}^{m} K(\boldsymbol{H}^{-\frac{1}{2}}(\boldsymbol{y} - \boldsymbol{x}_i)), \tag{42}$$

where $\boldsymbol{H}$ is the d-dimensional bandwith matrix. The accuracy of multivariate kernel density estimation is strongly dependent on $\boldsymbol{H}$ but only weakly on the choice of K [29]. As for the univariate case, we can choose the Gaussian kernel for the multivariate case too. This kernel is given by

$$K(\boldsymbol{u}) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\boldsymbol{u}^T\boldsymbol{u}}. \tag{43}$$

The `ks` package in R provides an implementation of kernel density estimation for 1-dimensional to 6-dimensional data, namely the function `kde`. It automatically takes care of optimal selection of $\boldsymbol{H}$ (h in the univariate case) by fitting it to the data prior to actual kernel density estimation. The multivariate Gaussian kernel is used by default. Hence, `kde` is a suitable tool for calculation of multivariate LogS, by setting $f(\boldsymbol{y}) = \hat{f}_{\boldsymbol{H}}(\boldsymbol{y})$ in Eq. (9). A possible R implementation of LogS is then simply:

```
logs <- function(x,y){

  # Estimate kernel density and evaluate fhat(y)
  fhat_y <- kde(x,eval.points = y)$estimate

  # Log score
  -log(fhat_y)

}
```

Numerical estimation of the k-variate PDF with `kde` is based on a discrete grid of g^k cells. The resolution of the grid naturally affects the precision of the estimate. Thus, if one wishes to have the same precision for any dimension, g should be kept constant and the time complexity of LogS is then $\mathcal{O}(g^k)$, i.e. exponential w.r.t. dimension. By default, `kde` happens to estimate 3-dimensional densities faster than 2-dimensional densities, indicating that the precision of the former might be suboptimal. However, this is not a vital issue for the message of this paper.

3.2. CRPS

Since CRPS is based on the conditional CDF, this CDF needs to be estimated. This can be obtained using the *empirical* CDF

$$\hat{F}(\boldsymbol{y}) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(\boldsymbol{x}_i \leq \boldsymbol{y}). \tag{44}$$

This estimator is chosen due to its simplicity and the speed with which it can be calculated numerically. Alternatively, a smooth kernel estimator that converges faster than the empirical CDF in terms of sample size is available in the literature [30]. A third option would be to integrate the kernel density estimate (Eq. (42)) discussed in Section 3.1,

$$\hat{F}_{\boldsymbol{H}}(\boldsymbol{y}) = \int_{-\infty}^{\boldsymbol{y}} \hat{f}_{\boldsymbol{H}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}, \tag{45}$$

which requires a numerical multivariate integration method at hand (see below).

After estimating the CDF, the CRPS can be evaluated as the sum of two integrals, as stated in Eq. (14). A proper implementation of numerical multivariate integration must be chosen with care in order to deal with the 'curse of dimensionality', which is neither trivially nor easily obtained. For the studies in this paper, we have chosen `vegas` from the `R2Cuba` package, which uses importance sampling [31]. It also automatically reports a measure of uncertainty. Alternatively, the `cubature` package offers adaptive multivariate integration, but from our experience it is slow compared to `vegas`. An R implementation of CRPS may be structured as follows,

```
crps <- function(x,y){

  # Bounds and dimensions
  l <- apply(x,2,min)
  u <- apply(x,2,max)
  m <- dim(x)[1]
  k <- dim(x)[2]

  # Estimate CDF
  Fhat <- defineECDF(x,m,k)

  # Lower integrand
  igdL <- function(u){
    Fhat(u)^2
  }

  # Upper integrand
  igdU <- function(u){
    (Fhat(u)-1)^2
  }

  # Lower integral
  intL <- intCrps(k,igdL,l,y)

  # Upper integral
  intU <- intCrps(k,igdU,y,u)

  # CRPS
  intL$value + intU$value

}
```

where the `vegas` integration is embedded in a custom function `intCrps` to maintain clarity in the `crps` function. Further relevant code is available on Github.[2] For univariate evaluation with CRPS, there is no need for multivariate integration and this simplifies the implementation substantially. An alternative way of evaluating CRPS, where the CRPS of an ensemble is shown to be equal to a weighted sum of quantile scores, has been proposed [32]. Thus the integration operation is effectively avoided. However, this approach is only described for univariate scores, and no attempt on generalization to multivariate CRPS is known.

3.3. Variogram score

For VarS, the forecast variogram must be estimated for each unique pair of horizons, i and j, cf. Section 2.1.3. That is equivalent to the expected pairwise difference to the power p, $E[|X_i - X_j|^p]$. Given an ensemble of the form in Eq. (37), this can be estimated as an average,

$$E[|X_i - X_j|^p] \approx \frac{1}{m}\sum_{l=1}^{m} |x_i^{(l)} - x_j^{(l)}|^p. \tag{46}$$

VarS can then be calculated using Eq. (20), for example with the following R implementation:

```
varP <- function(x,y,p=0.5){

  m <- dim(x)[1] # Size of ensemble
  k <- dim(x)[2] # Maximal forecast horizon

  # Iterate through all pairs
  score <- 0
  for(i in 1:(k-1)){
    for(j in (i+1):k){

      Ediff <- 1/m*sum(abs(x[,i]-x[,j])^p)
      score <- score + (abs(y[i]-y[j])^p - Ediff)^2

    }
  }

  # Variogram score
  return(score)

}
```

For this implementation, the time complexity of VarS is $\mathcal{O}(mk^2)$, i.e. quadratic w.r.t. the dimension k of the forecast and linear w.r.t. the ensemble size m. The computation of every single VarS is furthermore made from simple and cheap arithmetic operations and therefore it is reasonable to consider VarS as a computationally fast and scalable scoring rule compared to multivariate LogS and CRPS.

4. Simulation study

In this section, we construct a simulation study in which we apply various multivariate forecasts with different characteristics to a time series of simulated observations. The aim is to highlight the strengths and weaknesses of the scoring rules previously discussed.

We want to forecast a k-dimensional random variable that is multivariate by means of temporal correlation. Hence, we use the setup from Section 3, i.e. we aim to forecast

$$\boldsymbol{Y}_t = (Y_{t+1}, \cdots, Y_{t+k})^\top. \tag{47}$$

All probabilistic forecasts will be based on ensembles constructed as in Eq. (37). All displayed prediction intervals are 95%-prediction intervals. The relevant data are available on Github[3].

4.1. Case study 1 – a simple autoregressive model

To begin with, consider a simple autoregressive model of order 1 (AR(1)),

$$X_t = \phi X_{t-1} + \varepsilon_t. \tag{48}$$

[2] https://github.com/matbbDTU/probforecasteval.

[3] https://github.com/matbbDTU/probforecasteval.

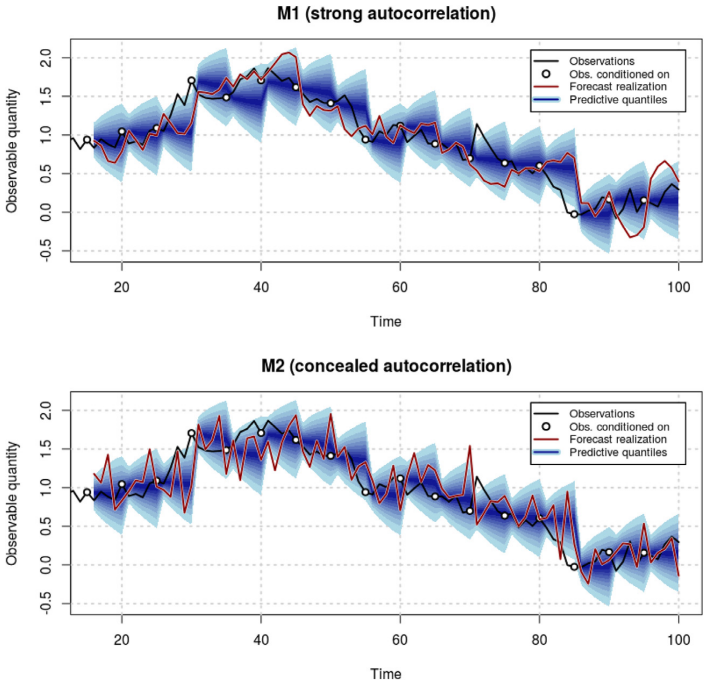

Fig. 8. 5-dimensional (w.r.t. forecast horizon) probabilistic forecasts of $\boldsymbol{y}_t^{(5)}$ generated by simulating $m = 2000$ realizations of Eq. (49). The probabilistic forecasts shown here are issued every 5th time step (observations conditioned on are highlighted in white). Autocorrelation is altered between the two models M1 and M2 by tuning the weights on the two noise parameters, σ_ε and σ_e, cf. the model overview above. **Top**: model M1 with strong autocorrelation (sample autocorrelation = 0.91); **Bottom**: model M2 with weaker autocorrelation (sample autocorrelation = 0.79). The difference in autocorrelation is indicated by displaying one random realization of each forecast model (dark red). The marginal distributions are identical between the two models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\varepsilon_t \sim$ i.i.d $\mathcal{N}(0, \sigma_\varepsilon^2)$. By choosing $\phi = 0.99$, we have a stochastic process with strong autocorrelation. An example of a simulated realization along with 5-dimensional ensemble forecasts is shown in Fig. 8 (top). The forecast series has a "sawy" look because the 5-dimensional forecasts are issued only at every fifth time step.

We now want to construct a forecast that preserves all its marginal distributions, but with altered correlation structure, as illustrated in Fig. 8 (bottom). Therefore, we introduce an observation equation

$$Y_t = X_t + e_t, \tag{49}$$

that adds observation noise, with $e_t \sim$ i.i.d $\mathcal{N}(0, \sigma_e^2)$ and e_t and ε_t mutually independent. Y_t is then effectively an autoregressive-moving-average model of order 1 (ARMA(1,1)). It follows that the total variance of Y_t is

$$\mathrm{Var}[Y_t] = \sigma_e^2 + \frac{\sigma_\varepsilon^2}{1-\phi^2}. \tag{50}$$

Adding more observation noise, i.e. increasing σ_e will lower autocorrelation, and by tuning σ_e accordingly such that $\mathrm{Var}[Y_t]$ is unchanged, we have obtained what we desired.

For forecasting purposes, however, it is necessary to instead consider the conditional variance, which for the k'th horizon is

$$\mathrm{Var}[Y_{t+k|t}] = \sigma_e^2 + \sigma_\varepsilon^2 \Big(\sum_{i=1}^{k} \phi^{2(i-1)}\Big). \tag{51}$$

To initiate the simulation study, let $\phi = 0.99$, unconditional $\mathrm{Var}[Y_t] = 1$, and simulate one realization $y = (y_1, y_2, \cdots, y_N)$ of $N = 100$ observations. Then, we consider three different prediction models, all with $\phi = 0.99$ and $\mathrm{Var}[Y_t] = 1$,

- **Model M1**: The true AR(1) model with $\sigma_e = 0$, i.e. $Y_t = X_t$.
- **Model M2**: Similar to M1, except with $\sigma_e = 0.99$. Since the total variance of 1 is preserved, σ_ε is lowered compared to M1.
- **Model M3**: Identical to M1 apart from the addition of a constant $\mu = 0.3$, i.e. $Y_t = X_t + \mu$.

Table 4
Computation time in seconds per 10,000 score evaluations.

Dimension	LogS	VarS
1	895.51	0.45
2	697.77	0.86
3	1819.81	1.45
4	6450.73	2.27

For each of the three models, we simulate ensembles of 2-dimensional to 5-dimensional predictive distributions yielding a total of 12 forecast series. Each series consists of $m = 100$ ensemble members. The 5-dimensional series are illustrated for models M1 and M2 in Fig. 8 (in this particular figure, $m = 20,000$ for smoother predictive quantiles), while M3 is a trivial variant of M1 and thus not displayed. The sample 1-step correlations for M1 and M2 were found to be 0.91 and 0.79, respectively.

4.1.1. Forecast evaluation

Each series is evaluated by all scoring rules considered in this article. For LogS and CRPS, both univariate and multivariate versions are applied to investigate the significance of the gain associated with upgrading from univariate to multivariate scoring rules. Since the marginal distributions of M1 and M2 are identical, the two models are expected to be deemed identical by univariate scoring rules, but separated by sufficiently effective multivariate scoring rules.

The results of the AR(1) simulation study are visualized in Fig. 9 (numbers can be found in Table 9). The following is observed for the 5 scoring methods,

1. Univariate LogS: M1 and M2 are equal as expected, M3 is inferior.
2. Multivariate LogS: M1 is better than M2. This is not apparent for $d = 2$ in the boxplot, but the difference can be verified by consulting Table 9 in Appendix. M3 is inferior.

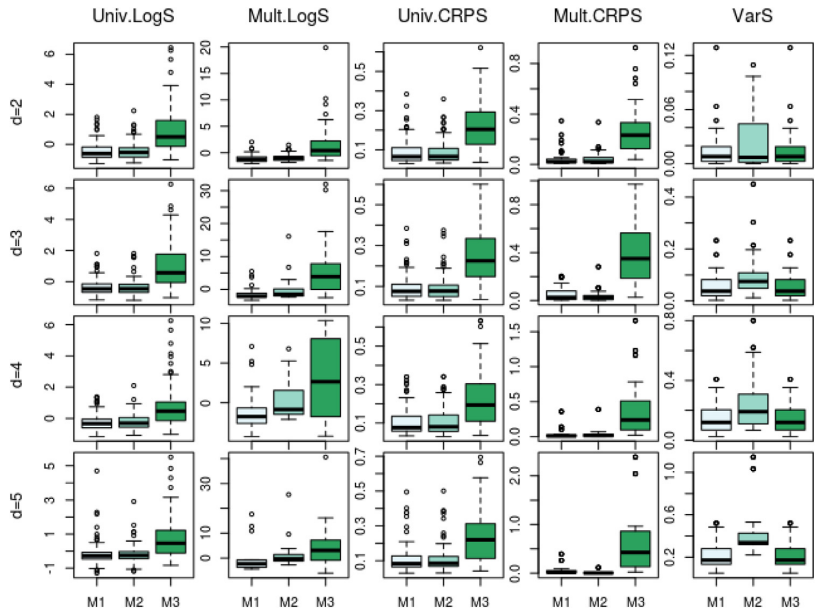

Fig. 9. Boxplots of scores resulting from the evaluation of M1, M2 and M3. Rows represent forecast dimensions, columns represent scoring rules applied.

3. Univariate CRPS: M1 and M2 are equal as expected, M3 is inferior.
4. Multivariate CRPS: M1 and M2 are equal, despite being evaluated by a multivariate scoring rule. M3 is inferior.
5. VarS: M1 is better than M2, but equal to M3.

Based on these observations, we can conclude:

- All LogS and CRPS measures agree that M3 is much worse than M1 and M2, across all dimension. M1 and M2 are generally not found to be different by LogS nor CRPS, except for higher-dimensional LogS.
- Going from univariate to multivariate LogS increases the ability to separate M1 and M2, i.e. the correlation structure is evaluated. This becomes more apparent for increasing dimension.
- Going from univariate to multivariate CRPS does not change any conclusions significantly, i.e. the correlation structure is not properly evaluated.
- VarS separates M1 and M2 better than LogS and CRPS, but the miscalibrated M3 is scored equally with M1.

Overall, for multivariate forecast evaluation, case study No. 1 suggests using either solely multivariate LogS, or VarS accompanied by univariate LogS or CRPS.

4.1.2. Run time comparisons

Based on the conclusions above alone, it seems tempting to always apply the multivariate LogS instead of having to evaluate the marginal densities with one scoring rule and the multivariate contribution with another. However, as previously stated, VarS is in theory much faster than LogS, especially with increasing dimension. This difference may be of relevance to the forecaster, depending on the scale of the problem at hand. Hence, timings of the scores evaluated above are reported and compared in Table 4 and Fig. 10. Since CRPS has now been shown to be useless for multivariate problems, it is not included here. For the 2-dimensional problem, VarS is about 2,000-times faster than LogS. Furthermore, in Fig. 10, the exploding behavior of LogS w.r.t dimension is obvious when compared to the moderately increasing run time of VarS, as expected. However the increment in time complexity for LogS is still less dramatic than expected, which means that the resolution has been reduced with increasing dimension, which again implies less reliable

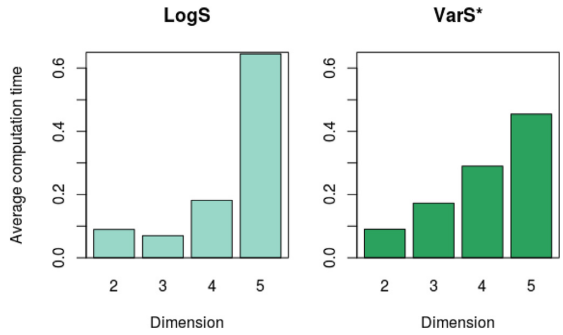

Fig. 10. Average computation time per score (*per 2000 scores for VarS to align scales).

estimates. The lesson learned is that there is an advantage in computational efficiency associated with evaluating multivariate forecasts with VarS instead of LogS, possibly already at 2 dimensions, dependning on the problem of concern and the resources available.

4.2. Case study 2 – a bounded point forecast-driven SDE model

In order to demonstrate the application of scoring rules on a case that resembles wind power production, we now generate a new series of observations, $\{y_t\}$, this time by simulating from a stochastic differential equation (SDE). This is a common modelling choice for wind power forecasting [33] as well as for solar power [34]. We choose the following SDE as the generating process

$$\mathrm{d}Y_t = \theta(\mu_t - Y_t)\mathrm{d}t + \sigma Y_t(1 - Y_t)\mathrm{d}W_t, \tag{52}$$

where W_t is the Wiener process [35]. The purpose of using this equation is to be able to simulate a quantity that resembles normalized wind power. This is in the sense that Eq. (52) has support on the interval (0,1) and the variance is dependent on the distance from y_t to the clos-

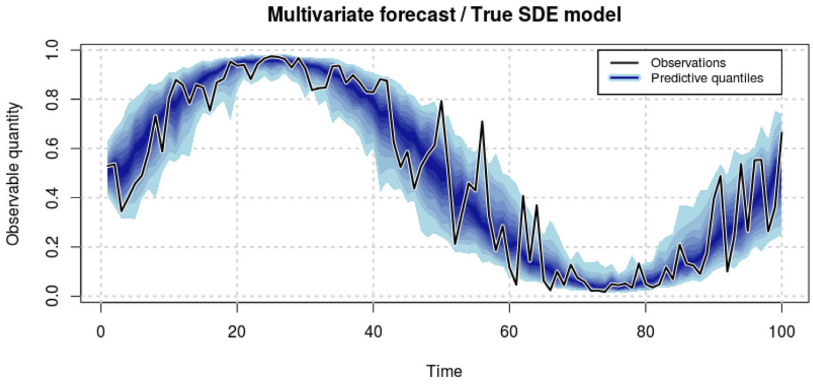

Fig. 11. 100-dimensional (w.r.t. forecast horizon) predictive distribution generated from Eq. (52). Notice the sine structure which appears due to the process being drived by the sine-based μ_t – a characteristic known from wind power prediction tools from the real world.

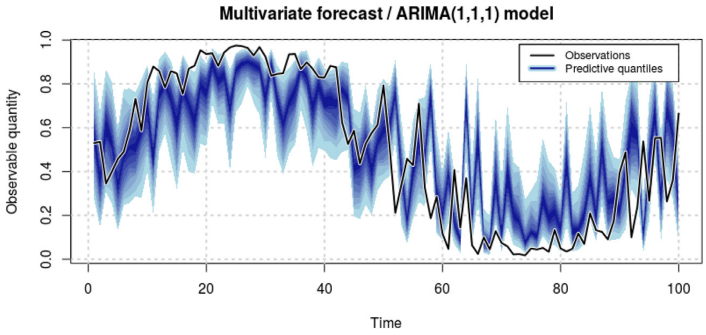

Fig. 12. 100 1-step predictive distributions generated by estimating Eq. (54) to fit y_t originally generated from Eq. (52).

Table 5
Scores comparing the two models Eqs. (52) and (54). All three scores separate the two competing models correctly in this case.

Model	LogS	CRPS	VarS
True SDE	1.217	0.052	84.560
ARIMA(1,1,1)	0.085	0.120	237.955

est boundary. μ_t represents a point forecast for which knowledge about its distribution is desired. Thus, we let

$$\mu_t = 0.45 \sin\left(\frac{2\pi}{100} t\right) + 0.5, \tag{53}$$

$\theta = 0.1$, and $\sigma = 0.3$ and simulate $\{y_t\}$. The realization is shown in Fig. 11, where the state-dependent variance is noticable - largest around $y_t = 0.5$, and smallest close to the boundaries.

We shall fit a competing autoregressive-integrated-moving-average model of order 1 (ARIMA(1,1,1)) $Z_t = Y_t - Y_{t-1}$ to y_t

$$Z_t = \phi Z_{t-1} + \theta \epsilon_{t-1} + \epsilon_t. \tag{54}$$

We choose to estimate the parameters of (54) in a rolling window, such that they are updated at every time step. In each time step, a one-step predictive distribution is generated; this is shown in Fig. 12.

Forecast evaluation

For this case, we shall apply VarS to the 100-dimensional forecast, and LogS as well as CRPS to the marginal forecast distributions. The results are summarized in Table 5 and illustrated in Fig. 13. Overall, we find that all three scores identify the correct model. Both LogS and CRPS identify the correct model in 78 out of 100 cases. VarS finds a seemingly huge difference between the two models, but it should be kept in mind that this is without any measure of uncertainty. The main purpose of case study No. 2 is to set the stage for case study No. 3, a more advanced and large-scale version of this case study.

4.3. Case study 3 – Klim Wind Power Plant

In the final case study, we shall examine a real data set, namely normalized wind power (w.r.t. maximum capacity) data from the Danish wind power plant, Klim. The data set consists of:

- Observations: Hourly average normalized wind power production, x_t.
- Predictions: Multivariate 48-h forecasts, $\tilde{\boldsymbol{p}}_t = (\tilde{p}_{t+1|t}, \tilde{p}_{t+2|t}, \ldots, \tilde{p}_{t+48|t})$ issued every six hours. The prediction horizon is denoted k.
- In total, there are 15,558 observations that can both be predicted and evaluated.

Similar to the SDE model in case study no. 2, where the point forecast μ_t drives the forecast density, we are going to simulate predictive distributions that depend on $\tilde{\boldsymbol{p}}_t$. We will use an already published SDE-model for forecasting [36], given by

$$\mathrm{d}Y_t = -\theta(Y_t - \tilde{p}_t - c\tilde{p}_t(1-\tilde{p}_t)(1-2Y_t))\mathrm{d}t + 2\sqrt{\theta\alpha\tilde{p}_t(1-\tilde{p}_t)}\mathrm{d}W_t, \tag{55}$$

where θ, $c \geq 0$ and $\alpha \in [0, 1)$ are constant parameters.

By simulating 300 realizations from Eq. (55), we obtain empirical 6-dimensional predictive distributions. The choice of horizon $k = 6$ is natural due to new observations coming in every 6th step (hour). A subset of the observations $\{y_t\}$ with $t = [100; 180]$ is shown along with the corresponding known point forecasts $\tilde{\boldsymbol{p}}_t$, and the 6-dimensional predictive distributions are shown in Fig. 14. The marginal predictive distributions of Y_t are generally skewed.

Forecast evaluation

With 15,558 marginal distributions to evaluate, computation time starts becoming a concern. We could use the same techniques as de-

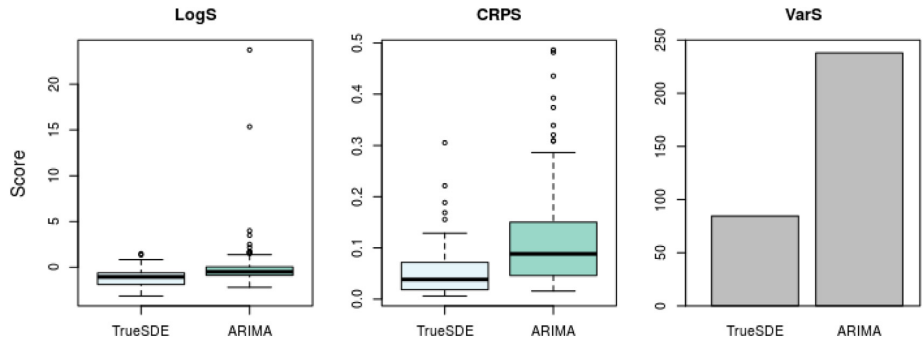

Fig. 13. Scores comparing the two models Eqs. (52) and (54). From the boxplots in the left and middle section, it is seen that LogS and CRPS, respectively, are both able to separate the two models correctly, although the distributions of scores are by no means completely distinct. VarS also clearly identifies the correct model with a score three-times lower than the wrong model.

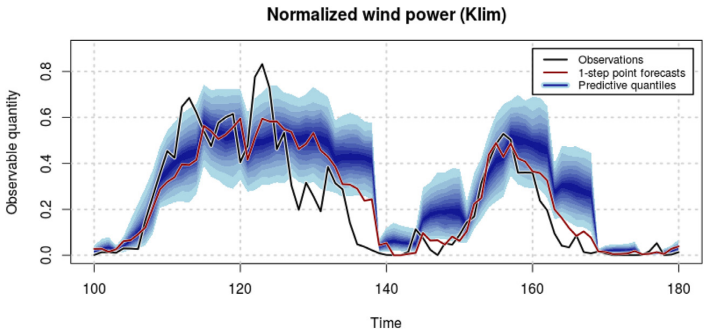

Fig. 14. Subset of the Klim normalized wind power data. Observations (white) as well as point forecasts issued by Klim (orange) along with 6-dimensional predictive distributions generated from Eq. (55) are displayed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

scribed above, although non-parametric CRPS in particular can be tricky to evaluate fast. Therefore, in this example we shall approximate each marginal predictive distribution by a parametric density and evaluate the difference.

Since y_t is double-bounded (0,1) and has a skewed distribution, the beta distribution is an obvious choice. Hence, we assume

$$Y_t \sim \text{Beta}(\alpha, \beta), \tag{56}$$

where α and β are shape parameters that completely characterize the beta distribution and that will be fitted uniquely to each of the 15,558 marginal distributions using maximum likelihood estimation.

By applying both the non-parametric (kernel density estimate) and parametric (beta density estimate) approximation, we get two models that seem to capture the characteristics of the marginal distributions very well, cf. the example in Fig. 15, which features the predictive density at time $t = 1025$. This example is representative for the rest of the marginal predictive densities.

Note that, even though the overall shape of the distribution in Fig. 15 is well-approximated by both models, there is a remarkable difference in the 0-end of the tail. Due to the characteristics of the beta distribution, it is not possible to have a bell-shaped density that is at the same time non-zero at $y_t = 0$, which is why we have $f(0) = 0$ for the parametric density. Nevertheless, 0-observations occur frequently for wind power, which is nicely captured by the non-parametric kernel density estimate.

For evaluation using LogS, the 0-probability creates a problem, since $\log(y)$ has a singularity in 0. The immediate solution is to add a very small number ν to all 0-observations. The challenge is that the choice of ν affects LogS greatly and thus creates potential for user-bias. Another possibility is to robustify the forecast density e.g. using Huber robustification [37], however then LogS is no longer proper, and this has to be taken into consideration.

Table 6
Average LogS and CRPS for the two models in concern.

Model	LogS	CRPS
Kernel (non-parametric)	11.648	0.0644
Beta (parametric)	63.511	0.0643

Table 7
Computation time in seconds per 10,000 score evaluations.

Model	LogS	CRPS
Kernel (non-parametric)	0.363	17100
Beta (parametric)	5.59e−3	2.297

With the important properties of VarS as well as multivariate vs. univarite LogS and CRPS well-covered in the first two cases, we shall just apply univariate LogS and CRPS for the final case. The results are summarized in Table 6 and visualized in Fig. 16.

The main observations are the following,

- The two models are practically identical according to CRPS
- The two models are significantly different according to LogS

The reason for this is the difference in probability mass in the tails, as conjectured above.

This case serves a second purpose, namely to examine the difference of computation speed between parametric and non-parametric models. Average running times of the computations above were thus estimated where each score evaluation was repeated 10 times. The results are shown in Table 7 in seconds of computation time per 10,000 scores.

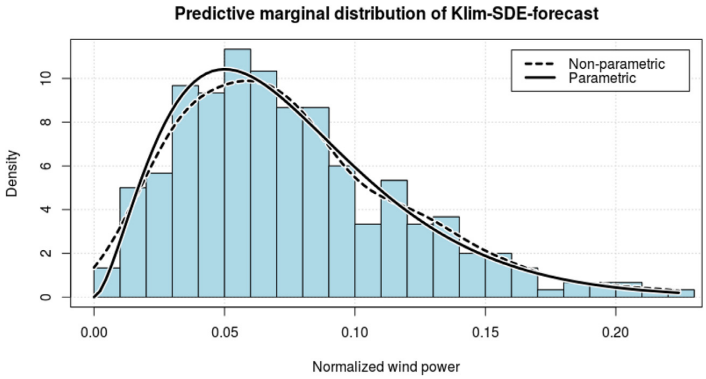

Fig. 15. Predictive marginal distribution for a random point in time (here $t = 1025$). A non-parametric (dashed line) as well as a parametric (solid line) approximation to the density are shown. Notice the difference in terms of probability mass in the left tail, while the overall shape is similar between the two models.

Table 8
Summary of the characteristics of LogS, CRPS and VarS revealed by the three case studies in Sections 4.1–4.3. The upper half of the table summarizes descriptive properties of the three scoring rules, where advantages are flagged with '+', and disadvantages are flagged with '÷'. The lower half summarizes the characteristics of the three scoring rules in verbal terms.

Advantages and disadvantages			
	LogS	CRPS	VarS
Calibration of mar-ginal distribution	+ Mean and variance are evaluated.	+ Mean and variance are evaluated.	÷ Mean is **not** evaluated.
Correlation structure	+ Fully evaluated.	÷ Very poorly evaluated, models can not be separated in practice.	+ Indirectly but effectively evaluated.
Run time at the k'th dimension	÷ Increases exponentially with k.	÷ Increases exponentially with k.	+ Increases quadratically with k.
Viability for multivariate problems	+/÷ Theoretically useful but too computationally demanding at higher dimension.	÷ Practically useless and computationally demanding.	+/÷ Partially useful and completely computationally feasible.
Summarized properties			
General characteristics	**LogS** Evaluates all information about the forecast distribution, but penalizes hard in the tails. Theoretically useful for any dimension, but not practically applicable at higher dimension.	**CRPS** Evaluates the overall shape of the forecast distribution including mean and variance, but fails to evaluate correlation in multivariate scenarios. Useful for one dimension, useless for multivariate problems.	**VarS** Evaluates correlation and variance indirectly, but fails to evaluate the mean. Partially useful for evaluation of multivariate forecasts but cannot stand alone.
Numerical methods required for implementation	Estimation of PDF, e.g. using a kernel density estimate, cf. Section 3.1.	1) Estimation of CDF, e.g. using the estimated CDF. 2) n-dimensional integration, e.g. using importance sampling, cf. Section 3.2.	Simple arithmetic operations, cf. Section 3.3.
Computational performance	Fast and accurate for lower dimension, computation time and memory footprint quickly increases with higher dimension.	Fast and accurate for lower dimension, computation time and memory footprint quickly increases with higher dimension.	Very fast for lower dimension, increase in run time and memory footprint is limited.

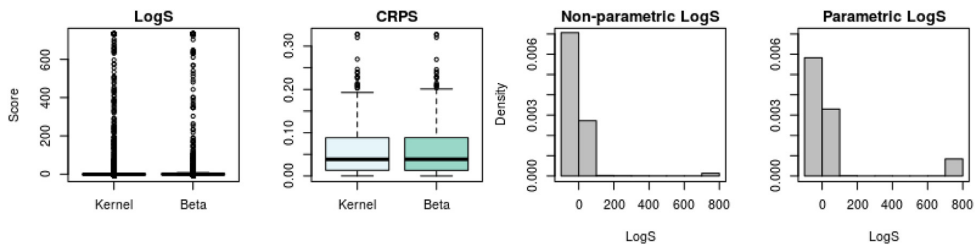

Fig. 16. LogS and CRPS evaluations of the non-parametric (kernel) and parametric (beta) model. The two barplots concern LogS only and are included to show that a larger proportion of the forecasts score badly in the parametric model, which is not possible to see from the boxplot.

Clearly, in the univariate case it is much faster to evaluate a parametric than a non-parametric distribution, especially for CRPS, where the difference is almost 1000-fold. Of course the computer in use as well as proper code optimization both play significant roles, and a more in-depth conclusion with regards to this issue requires a much more comprehensive study which is beyond the scope of this work.

4.4. Summary of characteristics of LogS, CRPS and VarS

Based on the application framework in Section 3 and the case studies in Section 4, we can summarize and compare interesting characteristics of the three scoring rules in question. These include calibration of the conditional expectation of a forecast, the ability to separate different

correlation structures, computation time required for non-parametric and parametric forecasts, respectively and finally scalability, i.e. the ability to maintain a reasonable computation time with an increasing number of dimensions. The findings are summarized in verbal terms in Table 8. Most importantly, it is clear that no scoring rule performs optimally at all aspects.

5. Conclusion

Probabilistic forecasts can be evaluated by applying different scoring rules, and we have discussed six, all applicable for multivariate forecasts, namely LogS, CRPS, VarS, DSS, CdL and CsL. The latter three are variants of LogS, hence only the former three have been thoroughly reviewed. To facilitate the bridge from formula to application, we have provided examples with basic calculations and suggestions for practical implementation in numerical scenarios.

We have constructed three case studies where we have applied LogS, CRPS and VarS to forecasts of different numbers of dimensions in order to highlight their advantages and drawbacks. Case study no. 1 serves to demonstrate the scoring rules' ability to separate competing forecasts in extreme cases. For multivariate problems, only multivariate LogS can handle calibration and correlation simultaneously. CRPS is sensitive to calibration but fails to detect misspecified correlation, while VarS excels at detecting misspecified correlation but fails to detect miscalibration. Despite LogS being superior in both disciplines, it also penalizes unlikely observations extremely hard compared to CRPS, which is clearly seen from the results in case study no. 3. This behavior may be adjusted by switching to CdL or CsL.

Regarding speed, VarS is very fast to compute, even for 100-dimensional problems, as seen in case study no. 2. Conversely, the computation times of both CRPS and LogS increase fairly dramatically with the number of dimensions. Case study no. 3 also shows that it is much faster to use a parametric approximation to the PDF or CDF than a non-parametric approximation, especially when computing CRPS.

Clearly, no scoring rule performs optimally at all aspects. In energy systems, both calibration, correlation and computation time are important, and in the general case it is therefore necessary to apply a combined evaluation approach rather than to apply only one of the scoring rules. Since multivariate evaluation is mainly of interest when the correlation structure of a forecast is assumed to be important, the multivariate part can be fully covered by applying VarS. However, in energy systems it is always crucial to have well-calibrated forecasts, and this can sufficiently be handled by applying univariate LogS or CRPS to all the marginal densities. This combined approach is demonstrated in case study no. 2 and ensures a fast computation time. The one question remaining for future studies is therefore precisely how a combination of VarS and one of the univariate scores in the best way can be formulated as one unified scoring rule, depending on the forecasting problem at hand.

Our overall recommendation for evaluation of a multivariate probabilistic forecast is thus to apply VarS to the full, multivariate forecast, while simultaneously evaluating its marginal densities by either univariate CRPS or LogS, depending on whether the shapes of the tails are considered important (LogS) or not (CRPS).

Declaration of Competing Interest

We have no conflicts of interest.

Acknowledgments

The work has been funded by the International Energy Agency and the Technical University of Denmark, and has partly been supported by the Centre for IT-Intelligent Energy Systems (CITIES) project funded by Innovation Fund Denmark under Grant no. 1305-00027B. We wish to thank Corinna Möhrlen (WEPROG), Niclas Brabrand Brok (DTU Compute) and Per Bækgaard (DTU Compute) for constructive criticism of the paper.

Appendix

Expectation of $|X_1 - X_2|^p$

Consider the random variable, $X = X_1 - X_2$, where

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}\right], \tag{57}$$

then

$$X \sim N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2 - 2\sigma_{12}). \tag{58}$$

In the special case where $\mu_1 = \mu_2$, the distribution of X is symmetric around $X = 0$, so we consider only $X > 0$ for a moment. Let $p = 0.5$, $g(x) = \sqrt{x}$ and $Y = g(X)$, i.e. $Y = X^p$. Because $g(x)$ is monotonic on $x > 0$, the change of variable principle can be applied,

$$\begin{aligned} f_Y &= \left|\tfrac{d}{dy}(g^{-1}(y))\right| \cdot f_X(g^{-1}(y)) \\ &= 2y \cdot f_X(y^2) \end{aligned} \tag{59}$$

.

Table 9
Case Study 1: average scores for the three competing models M1, M2 and M3.

Dim.	Score	M1 (Cor)	M2 (Unc)	M3 (Add)
$d = 2$	LogS (univ.)	0.4157	0.4196	0.9474
	LogS (multiv.)	1.0940	0.8574	1.6782
	CRPS (univ.)	0.0909	**0.0903**	0.2263
	CRPS (multiv.)	0.0499	**0.0434**	0.2751
	VarS	**0.0153**	0.0223	**0.0153**
$d = 3$	LogS (univ.)	0.3486	0.3462	1.0574
	LogS (multiv.)	1.232	0.1704	5.7756
	CRPS (univ.)	0.0954	**0.0949**	0.2399
	CRPS (multiv.)	0.0521	**0.0420**	0.4044
	VarS	**0.0584**	0.1015	**0.0584**
$d = 4$	LogS (univ.)	0.2413	0.2147	0.7879
	LogS (multiv.)	0.6793	0.3012	3.1261
	CRPS (univ.)	**0.1089**	0.1103	0.2252
	CRPS (multiv.)	**0.0402**	0.0419	0.4115
	VarS	**0.1485**	0.2588	**0.1485**
$d = 5$	LogS (univ.)	0.1398	0.1800	0.8038
	LogS (multiv.)	**0.3424**	1.7897	5.9828
	CRPS (univ.)	**0.1138**	0.1139	0.2306
	CRPS (multiv.)	0.0617	**0.0174**	0.6282
	VarS	**0.2330**	0.4374	**0.2330**

Because of the symmetry of f_X around $X = 0$, the true density of $|X_1 - X_2|^p$ is equal to $2f_Y$. Thus, the expectation of $|X_1 - X_2|^p$ is calculated from the usual expectation formula, which in this study in just evaluated numerically,

$$\begin{aligned}\mathrm{E}[|X|^p] &= \int_0^\infty u \cdot 2 f_Y(u) \mathrm{d}u \\ &= \int_0^\infty u \cdot 2 \cdot 2u \cdot \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2-2\sigma_{12})}} \mathrm{e}^{-\frac{(u^2-(\mu_1-\mu_2))^2}{2(\sigma_1^2+\sigma_2^2-2\sigma_{12})}} \mathrm{d}u \\ &= 4\int_0^\infty \frac{u^2}{\sqrt{2\pi\sigma_X^2}} \mathrm{e}^{-\frac{(u^2-\mu_X)^2}{2\sigma_X^2}} \mathrm{d}u.\end{aligned} \tag{60}$$

with $\mu_X = \mu_1 - \mu_2$ and $\sigma_X^2 = \sigma_1^2 + \sigma_2^2 - 2\sigma_{12}$.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.egyai.2021.100058

References

[1] Diebold FX, Lopez JA. 8 forecast evaluation and combination. HandbStat 1996;14:241–68.
[2] Gneiting T, Raftery AE. Strictly proper scoring rules, prediction, and estimation. J Am Stat Assoc 2007;102(477):359–78.
[3] Nielsen H, Nielsen T, Madsen H. An overview of wind power forecasts types and their use in large-scale integration of wind power. In: Proceedings of the 10th international workshop on large-scale integration of wind power into power systems; 2011. p. 25–6.
[4] Bessa RJ, Möhrlen C, Fundel V, Siefert M, Browell J, Haglund El Gaidi S, et al. Towards improved understanding of the applicability of uncertainty forecasts in the electric power industry. Energies 2017;10(9):1402.
[5] Palmer T. Towards the probabilistic earth-system simulator: a vision for the future of climate and weather prediction. Q J R Meteorol Soc 2012;138(665):841–61.
[6] Groen JJ, Paap R, Ravazzolo F. Real-time inflation forecasting in a changing world. J Bus Econ Stat 2013;31(1):29–44.
[7] Alkema L, Raftery AE, Clark SJ. Probabilistic projections of HIV prevalence using Bayesian melding. Ann Appl Stat 2007:229–48.
[8] Gneiting T, Katzfuss M. Probabilistic forecasting. Annu Rev Stat Appl 2014;1:125–51.
[9] Good IJ. Rational decisions. J R Stat Soc Ser B 1952:107–14.
[10] Matheson JE, Winkler RL. Scoring rules for continuous probability distributions. Manag Sci 1976;22(10):1087–96.
[11] Scheuerer M, Hamill TM. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. Mon Weather Rev 2015;143(4):1321–34.
[12] Diebold FX, Gunther TA, Tay AS. Evaluating density forecasts. 1997.
[13] Pinson P, Nielsen HA, Møller JK, Madsen H, Kariniotakis GN. Non-parametric probabilistic forecasts of wind power: required properties and evaluation. Wind Energy 2007;10(6):497–516.
[14] van der Meer D, Munkhammar J, Widén J. Probabilistic forecasting of solar power, electricity consumption and net load: Investigating the effect of seasons, aggregation and penetration on prediction intervals. Sol Energy 2018;171:397–413.
[15] Diebold FX, Hahn J, Tay AS. Multivariate density forecast evaluation and calibration in financial risk management: high-frequency returns on foreign exchange. Rev Econ Stat 1999;81(4):661–73.
[16] Clements MP, Smith J. Evaluating multivariate forecast densities: a comparison of two approaches. Int J Forecast 2002;18(3):397–407.
[17] Ko SI, Park SY. Multivariate density forecast evaluation: a modified approach. Int J Forecast 2013;29(3):431–41.
[18] Dovern J, Manner H. Robust evaluation of multivariate density forecasts2016.
[19] Gneiting T, Stanberry LI, Grimit EP, Held L, Johnson NA. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. Test 2008;17(2):211.
[20] Pinson P, Tastu J. Discrimination ability of the energy score2013.
[21] Dawid AP, Sebastiani P. Coherent dispersion criteria for optimal experimental design. Ann Stat 1999:65–81.
[22] Diks C, Panchenko V, Van Dijk D. Likelihood-based scoring rules for comparing density forecasts in tails. J Econ 2011;163(2):215–30.
[23] Schepen A, Everingham Y, Wang QJ. Coupling forecast calibration and data-driven downscaling for generating reliable, high-resolution, multivariate seasonal climate forecast ensembles at multiple sites. Int J Climatol 2020;40(4):2479–96.
[24] Schefzik R, Thorarinsdottir TL, Gneiting T, et al. Uncertainty quantification in complex simulation models using ensemble copula coupling. StatSci 2013;28(4):616–40.
[25] Nielsen HA, Madsen H, Nielsen TS, Badger J, Giebel G, Landberg L, et al. Wind power ensemble forecasting. In: Proceedings of the 2004 global windpower conference and exhibition; 2004.
[26] Madsen H. Time series analysis. Chapman and Hall/CRC; 2007.
[27] Silverman BW. Density estimation for statistics and data analysis, 26. CRC press; 1986.
[28] Epanechnikov VA. Non-parametric estimation of a multivariate probability density. Theory Probab Appl 1969;14(1):153–8.
[29] Wand MP, Jones MC. Comparison of smoothing parameterizations in bivariate kernel density estimation. J Am Stat Assoc 1993;88(422):520–8.
[30] Liu R, Yang L. Kernel estimation of multivariate cumulative distribution function. J Nonparametric Stat 2008;20(8):661–77.
[31] Ripley BD. Stochastic simulation. New York: John willey & sons; 1987.
[32] Bröcker J. Evaluating raw ensembles with the continuous ranked probability score. Q J R MeteorolSoc 2012;138(667):1611–17.
[33] Iversen EB, Morales JM, Møller JK, Trombe P-J, Madsen H. Leveraging stochastic differential equations for probabilistic forecasting of wind power using a dynamic power curve. Wind Energy 2017;20(1):33–44.
[34] Iversen EB, Morales JM, Møller JK, Madsen H. Probabilistic forecasts of solar irradiance using stochastic differential equations. Environmetrics 2014;25(3):152–64.
[35] Wiener N. Differential-space. J Math Phys 1923;2(1–4):131–74.
[36] Møller JK, Zugno M, Madsen H. Probabilistic forecasts of wind power generation by stochastic differential equation models. Journal of Forecasting 2016;35(3):189–205. doi:10.1002/for.2367. For.2367
[37] Windham MP. Robustifying model fitting. J R Stat Soc Ser B 1995:599–609.

Bibliography

Joseph Berkson. A statistically precise and relatively simple method of estimating the bio-assay with quantal response, based on the logistic function. *Journal of the American Statistical Association*, 48(263):565–599, 1953.

Keith J Beven and Michael J Kirkby. A physically based, variable contributing area model of basin hydrology/un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrological sciences journal*, 24 (1):43–69, 1979.

F Guillaume Blanchet, Pierre Legendre, and Daniel Borcard. Forward selection of explanatory variables. *Ecology*, 89(9):2623–2632, 2008.

Morten Borup. Real time updating in distributed urban rainfall runoff modelling. 2014.

Anders Breinholt, Fannar Örn Thordarson, Jan Kloppenborg Møller, Morten Grum, Peter Steen Mikkelsen, and Henrik Madsen. Grey-box modelling of flow in sewer systems with state-dependent diffusion. *Environmetrics*, 22(8): 946–961, 2011.

Anders Breinholt, Jan Kloppenborg Møller, Henrik Madsen, and Peter Steen Mikkelsen. A formal statistical approach to representing uncertainty in rainfall–runoff modelling with focus on residual analysis and probabilistic output evaluation–distinguishing simulation and prediction. *Journal of hydrology*, 472:36–52, 2012.

Niclas Laursen Brok, Henrik Madsen, and John Bagterp Jørgensen. Nonlinear model predictive control for stochastic differential equation systems. *IFAC-PapersOnLine*, 51(20):430–435, 2018.

Alexandre Costa, Antonio Crespo, Jorge Navarro, Gil Lizcano, Henrik Madsen, and Everaldo Feitosa. A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews*, 12(6): 1725–1744, 2008.

Hammouda Dakhlaoui, Zoubeida Bargaoui, and András Bárdossy. Toward a more efficient calibration schema for hbv rainfall–runoff model. *Journal of Hydrology*, 444:161–179, 2012.

DHI. *MOUSE Runoff Reference Manual*. DHI, Hørsholm, 2019. `https://manuals.mikepoweredbydhi.help/2019/Cities/MOUSERunoffReference.pdf`.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102 (477):359–378, 2007.

Irving John Good. Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 107–114, 1952.

Tao Hong, Pierre Pinson, Yi Wang, Rafał Weron, Dazhi Yang, and Hamidreza Zareipour. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*, 7:376–388, 2020.

Masaru Hoshiya and Etsuro Saito. Structural identification by extended kalman filter. *Journal of engineering mechanics*, 110(12):1757–1770, 1984.

Muhammad Jehanzaib, Muhammad Ajmal, Mohammed Achite, and Tae-Woong Kim. Comprehensive review: Advancements in rainfall-runoff modelling for flood mitigation. *Climate*, 10(10):147, 2022.

Rune Juhl, Niels Rode Kristensen, Peder Bacher, Jan Kloppenborg, and Henrik Madsen. Ctsm-r user guide. *Technical University of Denmark*, 2, 2013.

Niels Rode Kristensen, Henrik Madsen, and Sten Bay Jørgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, 2004.

Katarina Lavtar, Nejc Bezak, and Mojca Šraj. Rainfall-runoff modeling of the nested non-homogeneous sava river sub-catchments in slovenia. *Water*, 12(1): 128, 2019.

Thomas Lees, Marcus Buechel, Bailey Anderson, Louise Slater, Steven Reece, Gemma Coxon, and Simon J Dadson. Benchmarking data-driven rainfall-runoff models in great britain: A comparison of lstm-based models with four lumped conceptual models. *Hydrology and Earth System Sciences*, 25(10), 2021.

Roland Löwe, Søren Thorndahl, Peter Steen Mikkelsen, Michael R Rasmussen, and Henrik Madsen. Probabilistic online runoff forecasting for urban catchments using inputs from rain gauges as well as statically and dynamically adjusted weather radar. *Journal of Hydrology*, 512:397–407, 2014.

Roland Löwe, Luca Vezzaro, Peter Steen Mikkelsen, Morten Grum, and Henrik Madsen. Probabilistic runoff volume forecasting in risk-based optimization for rtc of urban drainage systems. *Environmental Modelling & Software*, 80: 143–158, 2016.

Roland Löwe, Rocco Palmitessa, Allan Peter Engsig-Karup, and Morten Grum. Fast and detailed emulation of urban drainage flows using physics-guided machine learning. In *EGU General Assembly Conference Abstracts*, pages EGU22–4303, 2022.

Nadia Schou Vorndran Lund, Anne Katrine Vinther Falk, Morten Borup, Henrik Madsen, and Peter Steen Mikkelsen. Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management. *Critical Reviews in Environmental Science and Technology*, 48 (3):279–339, 2018.

Henrik Madsen. *Time series analysis*. Chapman and Hall/CRC, 2007.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Surendra Kumar Mishra, Vijay P Singh, Surendra Kumar Mishra, and Vijay P Singh. Scs-cn method. *Soil conservation service curve number (SCS-CN) methodology*, pages 84–146, 2003.

Yusuf M Mohamoud and Lourdes M Prieto. Effect of temporal and spatial rainfall resolution on hspf predictive performance and parameter estimation. *Journal of Hydrologic Engineering*, 17(3):377–388, 2012.

Jan Kloppenborg Møller and Henrik Madsen. *From state dependent diffusion to constant diffusion in stochastic differential equations by the Lamperti transform*. DTU Informatics, 2010.

Susan A Murphy and Aad W Van der Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465, 2000.

Grey Stephen Nearing, Alden Keefe Sampson, Frederik Kratzert, and Jonathan Frame. Post-processing a conceptual rainfall-runoff model with an lstm. 2020.

Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

Umut Okkan, Zeynep Beril Ersoy, Ahmet Ali Kumanlioglu, and Okan Fistikoglu. Embedding machine learning techniques into a conceptual model to improve monthly runoff simulation: A nested hybrid rainfall-runoff modeling. *Journal of Hydrology*, 598:126433, 2021.

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations*, pages 65–84. Springer, 2003.

Rocco Palmitessa, Peter Steen Mikkelsen, Adrian WK Law, and Morten Borup. Data assimilation in hydrodynamic models for system-wide soft sensing and sensor validation for urban drainage tunnels. *Journal of Hydroinformatics*, 23(3):438–452, 2021.

John T Pedersen, John C Peters, and Otto J Helweg. Hydrographs by single linear reservoir model. Technical report, HYDROLOGIC ENGINEERING CENTER DAVIS CA, 1980.

Roberto Perin, Matteo Trigatti, Matteo Nicolini, Marina Campolo, and Daniele Goi. Automated calibration of the epa-swmm model for a small suburban catchment using pest: a case study. *Environmental Monitoring and Assessment*, 192:1–17, 2020.

Jim Pitman. *Probability*. Springer Science & Business Media, 1999.

Lewis A Rossman et al. *Storm water management model user's manual, version 5.0*. National Risk Management Research Laboratory, Office of Research and . . . , 2010.

Esteban Sañudo, Luis Cea, and Jerónimo Puertas. Modelling pluvial flooding in urban areas coupling the models iber and swmm. *Water*, 12(9):2647, 2020.

Michael Scheuerer and Thomas M Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

Jan Sitterson, Chris Knightes, Rajbir Parmar, Kurt Wolfe, Brian Avant, and Muluken Muche. An overview of rainfall-runoff model types. 2018.

Boni Su, Hong Huang, and Wei Zhu. An urban pluvial flood simulation model based on diffusive wave approximation of shallow water equations. *Hydrology Research*, 50(1):138–154, 2019.

Congcong Sun, Jan Lorenz Svensen, Morten Borup, Vicenç Puig, Gabriela Cembrano, and Luca Vezzaro. An mpc-enabled swmm implementation of the astlingen rtc benchmarking network. *Water*, 12(4):1034, 2020.

Christian Ankerstjerne Thilker, Rune Grønborg Junker, Peder Bacher, John Bagterp Jørgensen, and Henrik Madsen. Model predictive control based on stochastic grey-box models. In *Towards Energy Smart Homes*, pages 329–380. Springer, 2021.

Yazid Tikhamarine, Doudja Souag-Gamane, Ali Najah Ahmed, Saad Sh Sammen, Ozgur Kisi, Yuk Feng Huang, and Ahmed El-Shafie. Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs. particle swarm optimization. *Journal of Hydrology*, 589:125133, 2020.

Herbert JAF Tulleken. Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2):285–308, 1993.

Song Pham Van, Hoang Minh Le, Dat Vi Thanh, Thanh Duc Dang, Ho Huu Loc, and Duong Tran Anh. Deep learning convolutional neural network in rainfall–runoff modelling. *Journal of Hydroinformatics*, 22(3):541–561, 2020.

Norbert Wiener. Differential-space. *Journal of Mathematics and Physics*, 2 (1-4):131–174, 1923.

Frank Z Xing, Erik Cambria, and Roy E Welsch. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.

Kazuki Yokoo, Kei Ishida, Ali Ercan, Tongbi Tu, Takeyoshi Nagasato, Masato Kiyama, and Motoki Amagasaki. Capabilities of deep learning models on learning physical relationships: Case of rainfall-runoff modeling with lstm. *Science of The Total Environment*, 802:149876, 2022.

Gang Zhao, Zongxue Xu, Bo Pang, Tongbi Tu, Liyang Xu, and Longgang Du. An enhanced inundation method for urban flood hazard mapping at the large catchment scale. *Journal of Hydrology*, 571:873–882, 2019.

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.