A Bayesian Approach to Yield Curve Modelling and Forecasting with Stochastic Volatility for Interest Rate Risk Management

by

David Sarkisian

for the purpose of obtaining the degree of Master of Science
at the Delft University of Technology
to be defended publicly on Wednesday January 31st, 2024 at 10:30

Student number:	4547616	
Project duration:	March 1, 2023 - January 31, 2024	
Thesis committee:	dr. ir. J. Bierkens,	TU Delft, supervisor
	dr. ir. L. E. Meester,	TU Delft,
	drs. I. W. Specker,	DSTA, supervisor

TU Delft

Abstract

This thesis explores how forecasts of Dutch government bond yields can be improved by extending the current Dynamic Nelson-Siegel (DNS) model, used by the Dutch State Treasury Agency (DSTA), with stochastic volatility modeling and a Bayesian approach to parameter estimation and forecasting. The primary goal was to determine if the model extensions together with the Bayesian approach could improve the accuracy of yield forecasts given the highly volatile interest rate environment. In particular, we aimed to improve the "worst-case" forecasts, which we have defined as the upper bound of the 95% credible region with respect to the observed bond yields. To this end, we began with a baseline state-space model, resembling the current model in a state-space framework. Subsequently, we applied the findings from both in-sample and forecasting results as well as the findings from a literature review on volatility modeling to develop different models including two volatility models.

The volatility of the DNS model extensions is modeled as a GARCH process through the observation noise based on findings in the literature. This allowed for computationally efficient state estimation using a modified Kalman filter. Then, employing the Random Walk Metropolis algorithm for parameter estimation allowed us to use Bayesian multiple-step ahead forecasting. In particular, a comparative analysis of various models showed that while the current model performed better than expected, it was significantly outperformed in-sample by the DNS model with AR(1) observation noise (DNS-ARRW) and the DNS model with GARCH(1,1) observation noise volatility (DNS-OV). The Bayesian forecasting method particularly improved capturing the uncertainty of increasing yields in twelve-months ahead forecasts. Moreover, the two volatility models showed promising in-sample performance, but only one (DNS-OV) showed relatively good forecasting performance as well. Furthermore, the DNS-ARRW model consistently showed the best performance both in-sample and in forecasting.

In conclusion, the Bayesian approach to parameter estimation and forecasting proved effective in accounting for more variability in increasing forecast yields and simulating the direction of forecasts slightly better than the current MLE-based method. Moreover, the DNS-ARRW model showed significantly better worst-case forecasting performance, whereas the volatility models had a mixed performance.

Keywords: yield curve modeling, Bayesian forecasting, stochastic volatility, Dynamic Nelson-Siegel (DNS), Markov Chain Monte Carlo (MCMC)

“No matter how sophisticated our choices, how good we are at dominating the odds, randomness will have the last word.”

— Nassim N. Taleb, 2007

Preface

This thesis examines how modeling stochastic volatility in yield curve models together with using Bayesian forecasting and parameter estimation methods can improve the accuracy of yield forecasts. This thesis has been written to obtain the degree of Master of Science in Applied Mathematics. The research has been conducted under the supervision of Prof. Dr. Joris Bierkens in the Statistics department of Applied Mathematics.

In search of a Master's thesis project, I wanted to combine my passion for both applied mathematics and the public matter. Therefore, I am very grateful for the Dutch State Treasury Agency for offering me the opportunity to carry out this thesis project and combine my two greatest interests. Specifically, I want to give special thanks to my daily supervisor Ivo Specker for his continuing support, sharp feedback and for his approachability whenever I needed someone to brainstorm with throughout the process. Moreover, I would also like to express my special thanks to all of the colleagues in (but not limited to) the Policy and Risk Management department that contributed in making my internship the past eleven months a memorable experience. Furthermore, I would also like to thank my initial daily supervisor Marieke Grebe, who in the first few months contributed a lot of thought to my thesis as well.

I would especially like to express my gratitude to Prof. Dr. Joris Bierkens. I consider myself very lucky to have had a thesis supervisor that showed so much enthusiasm and engagement throughout this whole process. Our many discussions have contributed significantly in achieving the goals of this thesis and in my development as a mathematician in the last phase of my studies. Moreover, I would like to extend my gratitude to Dr. Ludolf Meester for taking seat in my thesis committee.

Furthermore, I would like to thank my close friends from high school and my student time for the many inspiring talks about mathematics, study and life. I want to thank my Laura, whose support has been invaluable to me for the past year. Finally, I deeply thank my parents for their unconditional love and support in both my academic and personal endeavours over the past many years.

David Sarkisian
Delft, January 2024

Contents

1

Introduction

In the aftermath of the COVID-19 pandemic, economies around the world have found themselves struggling with restarting due to supply chain disruptions and rising geopolitical tensions. These disruptions and tensions have led to higher production and energy costs, which stimulated inflation that reached levels not seen since the 70s and 80s. As a result, central banks have increased interest rates to tackle the high inflation. In particular, the European Central Bank (ECB) has increased the interest rate[1] 10 times from -0.50% in July 2022 to 4.00% in December 2023, which is a total increase of 450 basis points (bps) (ECB, 2023). The uncertainty that arises from potential interest rate hikes together with the overall uncertainty in the world results into more volatile bond markets, even relative to the stock market. In Figure 1.1 the VIX index and the MOVE index, two popular indicators for the volatility of the stock market and bond market (Kumar et al., 2022) respectively, are compared by looking at the VIX/MOVE ratio. Specifically, the volatile bond market (high MOVE) and relatively calm stock market (low VIX) result into a VIX/MOVE ratio that is historically low, indicating a relatively high volatile bond market.

Particularly, the highly volatile bond markets affect the Dutch state as well. The Dutch State Treasury Agency (DSTA) is responsible for managing the Dutch state debt and issuing government bonds. The issuance of bonds means that the Dutch state has costs arising from annual *coupon* (interest) payments and the payments of the principal amount of maturing bonds. Consequently, the DSTA knows the repayment schedule of all redemptions and coupon payments for bonds that have already been issued. However, interest rate costs in the future consist of known costs of already issued bonds and of unknown costs that arise from future bond issuance. The unknown interest rate costs are affected by the annual government budget deficit and the bond yields for different maturities in the market. Essentially, the annual government budget deficit influences the total amount of money that the DSTA has to raise from the market, mainly affecting the principal amounts due for repayment over the years. Additionally, the bond yield for some maturity observed in the market influences the coupon rate of a bond with a comparable maturity that has to be issued. However, the coupon rate is not determined by the exact bond yield for a comparable maturity,

[1] When talking about central banks increasing "the interest rate", we usually refer to the rate for which commercial banks and other financial institutes have to deposit some reserve funds overnight at the relevant central bank.

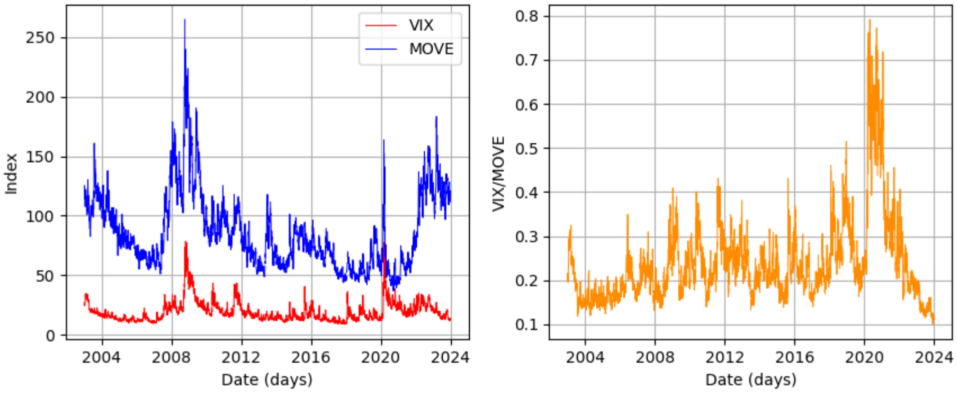


Figure 1.1: The VIX and MOVE indices from January 2nd 2003 till January 2nd 2024 (left) and the corresponding VIX/MOVE ratio (right).

but usually it is quite in the same range. Therefore, a forecast of the ten years bond yield is useful to forecast interest rate costs associated with bonds of a maturity of ten years. Then, a highly volatile interest rate environment complicates forecasting interest rate costs, as the interest rates show large deviations in a short period of time. Moreover, the estimated funding need of the Dutch state in 2024 is around €75 billion of which €40 billion is expected to be raised from the capital market, which are the bonds with a maturity longer than one year[2] (DSTA, 2023). Additionally, the estimated interest rate costs for 2024 are around €7.8 billion (Rijksoverheid, 2023). So, estimating interest rate costs is important for managing the interest rate risks arising from managing the state debt as it involves substantial amounts of money. The DSTA estimates future interest rate costs by forecasting interest rates and using the predicted interest rates to estimate the costs that arise from future issued bonds, whereas the funding need is estimated by expert judgement. Consequently, in this thesis we only focus on modeling and forecasting the bond yields, so the uncertainty arising from the government budget deficit is out of the scope of this thesis.

Several methods exist to model bond yields for different maturities, or the yield curve, but central banks and debt management offices often use variants of the Dynamic Nelson-Siegel (DNS) model (Filipovic, 2009, p. 3) introduced by Diebold et al. (2006), which is a state-space extension of the original Nelson-Siegel model of C. R. Nelson and Siegel (1987). The DNS model as state-space model assumes that the yield curve is driven by three underlying factors, called states. Particularly, Diebold and Li (2006) introduce an interpretation of the three states as the *level*, *slope* and *curvature* of a yield curve, which is commonly used nowadays to interpret the DNS model in an economical context. Moreover, Diebold and Li (2006) propose a two-step method characterized by the use of linear regressions to estimate the underlying factors as opposed to the one-step method proposed by Diebold et al. (2006) that takes advantage of the model being a state-space model and uses (extensions of) the Kalman filter to estimate the three states for every time point. Extensive

[2] The DSTA draws a distinction between the money market and the capital market. Bonds with a maturity of < 1 year are said to raise funds in the money market, whereas bonds with maturities > 1 year are said to raise funds in the capital market.

research has been done on the DNS model and various extensions have been proposed. In particular, there has been done research in modeling volatility in the DNS framework (Koopman et al., 2010, Hautsch and Ou, 2008a, Glosten et al., 1993, Mesters et al., 2014) that focus on adding volatility with either a GARCH process or a Stochastic Volatility process. These type of extensions usually require Bayesian techniques like Markov Chain Monte Carlo (MCMC) algorithms to estimate model parameters as the conventional Maximum Likelihood Estimation (MLE) methods struggle with finding optimal parameter values in higher dimensions due to the complex shape of such high-dimensional parameter spaces.

The current model and method employed by the DSTA is based on the two-step estimation method of Diebold and Li (2006) for the DNS model, using the ordinary least squares method to estimate states and an MLE-based method to estimate parameters. However, the current model and method have difficulty in predicting the increasing interest rates and the associated volatility due to interest rate hikes. Moreover, the current model and method do not quantify the uncertainty of the forecasts realistically. Specifically, the "worst-case" forecasts (upper bound of the 95% credible region of the forecast simulations) show quite some deviation from the actual interest rates. As a result, the forecasts of the current model have lost some practical significance. That is why this thesis aims to research whether extending the current model with stochastic volatility modeling and using Bayesian forecasting techniques can improve the interest rate forecasts. The main research question to this end is

- How to model interest rate volatility and quantify forecasting uncertainty with a Bayesian approach in order to forecast interest rate costs better?

In particular, we are interested in modeling bond yields for various maturities and, consequently, yield curves. The sub-questions that lead to answering the main research question are

1. How to model bond yields and how can we extend those models with stochastic volatility?
2. How can a Bayesian approach improve uncertainty quantification in yield curve forecasts?
3. How do the different models and methods compare?

In order to answer these questions, this thesis is written in the following structure. In Chapter 2 we provide some background information on interest rates and discuss the used data. Subsequently, in Chapter 3 we discuss the theory on state-space models, which are the used type of yield curve models. Then, in Chapter 4 we introduce the Bayesian methods that are used for parameter estimation and forecasting. In Chapter 5 we provide a literature review on yield curve and volatility modeling in the DNS framework. Afterwards, in Chapter 6 we bring the Bayesian approach together with yield curve modeling and we discuss the explored yield curve models. Then, in Chapter 7 we discuss the parameter estimation, in-sample and forecasting results for each model. Additionally, we compare the in-sample and forecasting performance of the used models and Bayesian methods with the current model and method in Section 7.6 to answer the main research question. Finally, we discuss the main conclusion, the limitations of our research and recommendations for further research in Chapter 8.

It is worth noting for the readability of this thesis that we have worked in a modeling cycle, which means that we have explored one model and we have used the findings of that model to explore a new model. So, the formulation and explanation of each model are provided in Chapter 6, whereas the results that lead to a certain model can be found in Chapter 7.

2

Data and Introduction to Bonds

In this chapter we introduce some key concepts related to bonds that will be used throughout this thesis and we discuss the used bond yield data as well. In Section 2.1 and 2.2 we introduce two common types of bonds. Subsequently, in Section 2.3 we discuss the concepts of yield-to-maturity and a yield curve for the previously introduced bonds. Then, we conclude this chapter by discussing the used bond yield data.

2.1 Zero-Coupon Bonds

Let us first consider the zero-coupon bond. This is a bond that has no cash flows between buying at some time t an maturity T. We provide a definition of the zero-coupon bond based on Oosterlee and Grzelak (2019, p. 341).

Definition 2.1. *(Zero-coupon bond). A zero-coupon bond (sometimes called a discount bond) is a contract with maturity T and at time $t \leq T$ value $P(t,T)$, which pays €1 at maturity T, denoted as $P(T,T) = 1$. The €1 that is paid at maturity is called the notional amount, sometimes also called the principal amount, and is more generally denoted by N.*

Since the only payment is the notional amount at maturity with certainty, the price of the zero-coupon bond $P(t,T)$ represents a *discount* on the notional amount N. The discount on the notional amount can be seen as a compensation for the time value of money (Berk et al., 2021, p. 162). Zero-coupon bonds typically have short-term maturities. For instance, Dutch Treasury Certificates or U.S. Treasury Bills with a maturity of less than one year are usually zero-coupon bonds.

2.2 Coupon Bonds

Coupon bonds are bonds that pay *coupon*, or interest, at fixed dates over the life of a bond. This means that there are cash flows between the issuance of the bond and the time of maturity. The coupon rate can be a fixed rate for all payments (fixed rate) or can be linked to some unknown future market rate (floating rate) like the euro short-term rate (ESTR). Since the DSTA only issues bonds with a fixed coupon rate we only focus on fixed-rate bonds. Oosterlee and Grzelak (2019, pp. 341-342) define a fixed-rate bond as follows.

Definition 2.2. *(Fixed-rate bond). For a given fixed rate r, a notional amount N and a set of payment dates $T_1, T_2, \ldots, T_m$, a fixed coupon rate bond is an investment with several coupon payments, that are defined by*

$$V_i(T_i) = \begin{cases} rN(T_i - T_{i-1}), & i = 1, 2, \ldots m-1 \\ rN(T_m - T_{m-1}) + N, & i = m. \end{cases} \tag{2.1}$$

Note that, due to the interim cash flows of a fixed-rate bond, the price of a fixed-rate bond cannot directly be considered as the compensation of the time value of money. Bonds with coupon payments typically have medium to long-term maturities. Examples include the Dutch State Loans or U.S. Treasury Notes with maturities ranging from one to thirty years.

2.3 Yield-to-Maturity and Yield Curve

In Section 2.1 we introduced the price of a zero-coupon bond, $P(t,T)$. However, in practice the bond price is not used directly, but rather the *yield-to-maturity* of a bond is used. This is the interest rate at which a bond is traded (Oosterlee and Grzelak, 2019, p. 375). The yield-to-maturity, or just *yield*, of a zero-coupon bond is given by (Berk et al., 2021, p. 163)

$$y(t,T) = \left(\frac{N}{P(t,T)}\right)^{\frac{1}{T-t}} - 1, \tag{2.2}$$

where N and $P(t,T)$ are again the notional amount and the price of a zero-coupon bond with maturity T at time t respectively. Since a zero-coupon bond does not have coupon payments, the yield-to-maturity for the zero-coupon bond is just the future cash flow at maturity that is discounted to the present value. In contrast, the yield for a fixed-rate bond cannot be expressed in closed-form since the additional cash flows of the coupon payments have to be taken into account as well, for which we refer to Berk et al. (2021, Section 6.3) for a more detailed discussion. This means that in general we have to use the zero-coupon yields or we have to discount the cash flows of coupon-bearing bonds that include coupon payments, since the zero-coupon yields are considered to represent the time value of money. So, for applications such as bond pricing, it is common practice to use the zero-coupon yields instead of the yields of a fixed-rate bond.

Subsequently, bond yields can vary across different maturities. The mapping $T \mapsto y(t,T)$ is commonly referred to as a yield curve. In theory, the yield curve is some smooth curve that relates the time-to-maturity $\tau = T - t$ with the bond yield $y(t,T)$ at some fixed time t. However, in reality bonds are not available for a continuous spectrum of maturities and we are dependent on market observations of the yields for various bonds. These are finite observations for maturities $\tau_1, \ldots, \tau_M$ that are possibly noisy (Filipovic, 2009, p. 29). This means that the actual yield curve at time t has to be estimated from different observations of bond yields at time t. In Figure 2.1 observations of actual Dutch government bond yields are shown at 5 different dates for maturities of 24, 36, 48, 60, 72, 84, 96, 108, 120, 240 and 360 months.

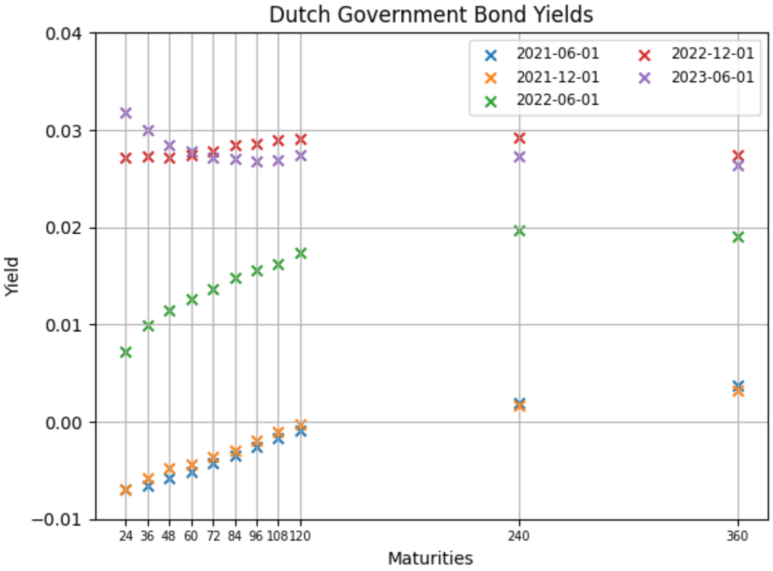


Figure 2.1: Dutch government bond yields as observed in the market at various dates.

2.4 Data

For this research we have used the bond yield data for eleven maturities $\tau \in \{24, 36, 48, 60, 72, 84, 96, 108, 120, 240, 360\}$ directly, which is shown in Figure 2.2. We note that the choice for these maturities is determined by the availability of certain ranges and the frequency of bond yields. So, we have to be cautious for the estimates of the bond yields of the short maturities ($<$ 24 months). In addition, we note that the observed bond yields of the medium to long-term maturities in the market are usually not zero-coupon bonds, but coupon-bearing bonds. This means that the yields also take into account the coupon payments and are not providing the zero-coupon bond yield curve, which is usually used for yield curve estimation.

In the literature so-called *unsmoothed Fama-Bliss zero-coupon bond yields*, which are modified U.S. Treasury rates, are often used to compare theoretical results between different papers. For the further details we refer to the original authors Fama and Bliss (1987). Another method to obtain zero-coupon yields is by using the so-called *bootstrap* method. This method involves iteratively computing the yields from the shortest-term to the longest-term maturity by "removing" the coupon payments from the yields, for which we refer to Berk et al. (2021, Section 6.3) and Filipovic (2009, Section 3.1).

However, we have chosen to use the direct bond yield data and not to modify these yields. The main reason for this is a practical one. In particular, the current model and method employed by the DSTA use the direct bond yield data as well. This is based on a paper of Ibáñez (2015) comparing a rigorous approach involving the bootstrap method with daily bond price data with a practitioner's approach involving constant maturity rates reported by the U.S. Treasury. Specifically, the author finds that the mean absolute error (MAE) values of the rigorous and practitioner's approach with respect to the actual yields are 8.97 bps and 8.92 bps respectively. Since the aim of this thesis is to explore models *with respect to* the current model and method it seems reasonable to use the same data to have a sound comparison between the different models and methods. So, we are aware of the somewhat practical approach to the used data. Naturally, this has both drawbacks and advantages.

First, the direct usage of the bond yields implies that the modeled and forecast yields in our research include the effects of the coupon payments. This means that the yields could overlook the price sensitivity (duration) of a bond as we do not consider the cash flows explicitly in the bond yields. Moreover, the used yields do not represent the time value of money directly as the zero-coupon bond yields. However, in reaching the goals of this research we are mainly interested in trends of the market yields as opposed to using the yields to price bonds or construct a portfolio strategy, for which the zero-coupon yields are needed. So, we argue that while direct yields are not suitable for bond pricing models or portfolio strategies, they are sufficient for forecasting the trends in bond yields.

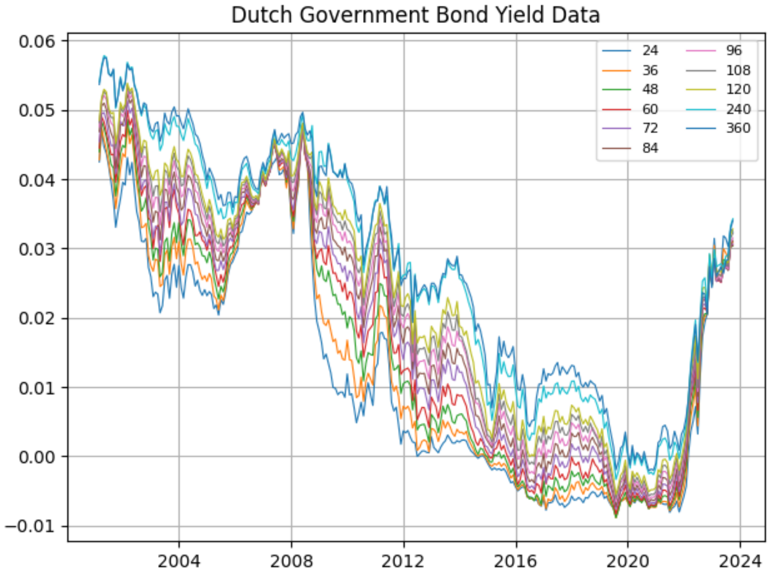


Figure 2.2: Graph of the used Dutch government bond yield data from March 2001 to October 2023 for maturities of 24, 36, 48, 60, 72, 84, 96, 108, 120, 240 and 360 months.

3

State-Space Models and State Estimation

In this chapter we will discuss *state-space models* (SSMs) and important subjects related to such models. In particular, we begin with explaining what state-space models are in Section 3.1. Subsequently, in Section 3.2 we will outline two properties of state-space models that are important for dealing with estimating states, which we will discuss in Section 3.3.

3.1 State-Space Models

Imagine that we want to know whether it rains or not, but we are stuck in some office without any windows. The only way we can tell whether it has rained is to look at whether employees are entering the office with or without an umbrella. One can imagine that looking at umbrellas is not the most exact way of observing whether it has rained or not. It could be the case that it rains often and people take their umbrellas just in case. On the contrary, it could also be the case that employees never take an umbrella with them to the office since it is a few seconds walk from the train station. In the context of state-space models, we call the situation whether it rains or not the *state*. We try to say something about the state by *observing* employees with or without umbrellas. We can extend this anecdote to a more specific example. Suppose that we want to know what the *true*, or exact, rainfall is. We call the true rainfall the **state**, in which we are interested. The true rainfall is something that we cannot know with a 100% certainty and has some randomness to it. Nature is random, so a region can be struck with an unforeseen heat wave or an unexpected storm. This uncertainty that influences the rainfall results into an uncertain - a "noisy" - state, and this randomness is called the **state noise**. Since the true rainfall is not directly observable we have to think of a method to measure rainfall and a way to link those measurements with the true rainfall.

We begin by putting an empty water tank outside every day. If it rains heavily on a given day, the tank will be filled up with water, whereas the tank will be empty on a dry day. We call the water level that is reached in the water tank an **observation**. One can imagine that these observations will not be very accurate. In particular, there could be small cracks in the tank that let go a little bit of the collected water or the measure lines on the tank could be slightly off, which contribute to less measurement accuracy. A digital rain gauge could help us obtaining more accurate measurements, but electronics are also prone to errors in their electronic circuits or there could be some production error. Most importantly, there is always some error (be it perhaps extremely small) in the measurements that results into uncertainty in the observation itself, which becomes a "noisy" observation. This is called the **observation noise**. Let us summarize the components thus far,

- state x_t: the *true* rainfall at day t;
- state noise v_t: the uncertainty in nature affecting the rainfall at day t;
- observation y_t: the observed rainfall in the water tank at day t;
- observation noise w_t: the uncertainty in the measurement at day t.

Now we have the individual pieces of a state-space model, but they are still not connected. A first assumption we can make is that we believe the rainfall of today is somehow connected with the rainfall of yesterday. This does not have to be a one-on-one influence, but intuitively the possibility of having a rainy day if it has been dry for the past thirty days is quite small, while it is instinctively larger if it has already rained for a week. How the rainfall of yesterday influences that of today, can be rephrased as how the state at time t *transitions* to the state at time $t+1$ and is called the **state-transition**. The last piece that has to be connected is how we link the actual rainfall to the observed water level in the water tank. We can, for instance, say that the water level we see in the tank is twice the actual rainfall or in general $g \in \mathbb{R}_{>0}$ times the actual rainfall. We now have the last components to define the example state-space model given by

$$y_t = gx_t + w_t, \tag{3.1}$$

$$x_t = fx_{t-1} + v_t, \tag{3.2}$$

where $f \in \mathbb{R}$ is the state-transition factor representing the amount of influence the rainfall on a previous day has on the next one and g is the observation factor as explained before. Additionally, we assume for this example that the randomness for both the observation and the state, the noise terms v_t, w_t, are normally distributed

$$v_t \sim \mathcal{N}(0, q^2),\ w_t \sim \mathcal{N}(0, r^2), \tag{3.3}$$

which is also referred to as Gaussian white noise in short.

3.1.1 Linear and Gaussian State-Space Models

Thus far, we have introduced the concept of a state-space model by a simple physical example. More generally, though, we can extend the rainfall example to a state-space model with even more states like rainfall, groundwater level, soil moisture, etc. and more observations given by a rain gauge, a soil moisture sensor, a submersible pressure sensor, etc. This results into a multidimensional state-space model, where the states and observations can also have some "cross-influence", or cross-dependence, on each other. This can be formally summarized in the following definition given by Brockwell and Davis (1991, pp. 463-464).

Definition 3.1. *(Linear Gaussian State-Space Model). A time-series model can be represented in linear state-space form. By this we mean that the series $\{\boldsymbol{y}_t, t = 0, 1, \dots\}$ satisfies the equation*

$$\boldsymbol{y}_t = G_t \boldsymbol{x}_t + \boldsymbol{w}_t, \quad t = 0, 1, \dots, \tag{3.4}$$

$$\boldsymbol{x}_t = F_t \boldsymbol{x}_{t-1} + \boldsymbol{v}_t, \quad t = 0, 1, \dots, \tag{3.5}$$

where $F_t \in \mathbb{R}^{n \times n}, G_t \in \mathbb{R}^{p \times n}, \boldsymbol{y}_t \in \mathbb{R}^p, \boldsymbol{x}_t \in \mathbb{R}^n$ for $t = 0, 1, \dots$. Moreover, $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are independent and distributed as

$$\begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{v}_t \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} R_t & S_t^T \\ S_t & Q_t \end{bmatrix}\right), \tag{3.6}$$

with $Q_t \in \mathbb{R}^{n \times n}, R_t \in \mathbb{R}^{p \times p}, S \in \mathbb{R}^{n \times p}$.

Notice that the state-space model is linear in the state, since we can see the observation $\boldsymbol{y}_t$ as a linear transformation (matrix multiplication plus a noise term) and that the state-space model is Gaussian as both noise terms are assumed to be Gaussian (normally distributed). Moreover, the state-space model in Definition 3.1 is multidimensional, so we work with matrices instead of one-dimensional factors. So, F is the state-transition matrix, G is the observation matrix and $\boldsymbol{y}_t, \boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{v}_t$ are conceptually the same as their one-dimensional counterparts. Additionally, the matrices G_t, F_t, R_t, Q_t, S_t are together commonly referred to as the *system matrices*.

Furthermore, in the used linear state-space models for modeling yield curves we assume time-invariance, i.e. $F_t \equiv F$ and $G_t \equiv G$, and time-invariant observation and state noise covariance matrices $R_t \equiv R$ and $Q_t \equiv Q$ respectively. For our example, this means that the influence of the rainfall at day t on the rainfall at day $t + 1$ does not change over time, and the way we link the actual rainfall with measurements of rainfall in the water tank stays the same over time as well. Additionally, we also assume that the observation noise $\boldsymbol{w}_t$ and state noise $\boldsymbol{v}_t$ are independent of each other, so $S_t \equiv S = O$, where $O \in \mathbb{R}^{n \times p}$ is the matrix with zero-only entries. Consequently, with these assumptions and Definition 3.1 some (in)dependencies arise, which are useful for derivations in later sections and chapters. In particular, we see that the model assumes that $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_{t-1}$ are conditionally independent given $\boldsymbol{x}_t$, which is denoted as

$$\boldsymbol{x}_{t+1} \perp \boldsymbol{x}_{t-1} | \boldsymbol{x}_t, \tag{3.7}$$

which in our example corresponds with the actual rainfall of tomorrow $\boldsymbol{x}_{t+1}$ being only dependent of the actual rainfall of today $\boldsymbol{x}_t$. Furthermore, the model assumes that $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ are conditionally independent as well, so

$$\boldsymbol{y}_t \perp \boldsymbol{x}_{t-1} | \boldsymbol{x}_t, \tag{3.8}$$

which means that our observation of the water level in the water tank today $\boldsymbol{y}_t$ only depends on the actual rainfall of today $\boldsymbol{x}_t$. Finally, the observations $\boldsymbol{y}_1, \dots, \boldsymbol{y}_t$ are dependent of each other via the states. Bishop (2006) elaborates on the (in)dependencies by means of a visual representation (a graph) of the state-space model as in Figure 3.1.

In Figure 3.1 the nodes represent states ($\boldsymbol{x}_t$) or observations ($\boldsymbol{y}_t$) at some time $t = 1, \dots, k$. Here, the arrows represent the dependency structure of the state-space model, where $\text{node}_1 \longrightarrow \text{node}_2$

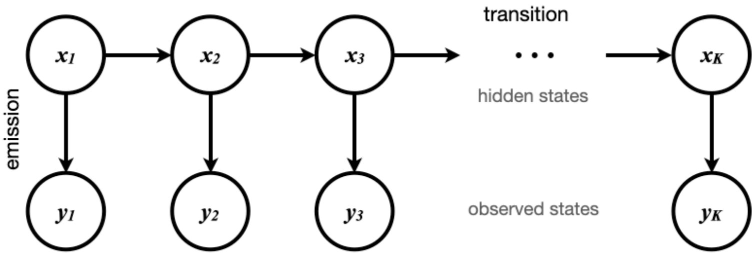


Figure 3.1: This is a schematic overview of a state-space model, where $\boldsymbol{z}$ is the state variable and $\boldsymbol{x}$ is the observation variable. (*Source*: Vivekvinushanth, 2020).

means node$_2$ is dependent on node$_1$. So, the first two dependence relations follow directly from the graph. The last dependence relation between the observations is a bit harder to see in the graph. Without going into too much graph theory (see Bishop, 2006, pp. 378-382 for more detailed theory), we can see that there is always a *path* from one observation to another, where a path is an edge regardless of the direction. In particular, consecutive observations always have a path between them that goes through consecutive states, on which the observations are dependent. So, instinctively the indirect dependence between the observations is something to take into account. Especially, when we need distributions like the posterior predictive distribution $p(\boldsymbol{y}_t|Y_{t-1})$, where $Y_{t-1} := \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}\}$, which is needed for forecasts and is discussed more thoroughly in Chapter 4.

3.1.2 Nonlinear and non-Gaussian State-Space Models

In the previous section we discussed linear Gaussian state-space models, which is the most basic type of state-space models. An obvious generalization is then to consider state-space models that are nonlinear or non-Gaussian. However, in this subsection we only discuss nonlinear state-space models, since we have not explored any non-Gaussian models. Then, we first have to specify what we exactly mean with "nonlinear". Nonlinear state-space models can be extensions that

1. are nonlinear in the state variable $\boldsymbol{x}_t$;
2. are nonlinear in the noise terms $\boldsymbol{w}_t$ or $\boldsymbol{v}_t$ (such as non-additive noise);
3. are nonlinear in the system matrices G_t, F_t, R_t, Q_t or S_t;
4. are some combination of the above.

Particularly, Harvey (1990, pp. 155-156) makes a distinction between the first type of nonlinearity in the state variable and the third type of nonlinearity in the system matrices. The author calls the first type *functionally nonlinear* and the third type *conditionally Gaussian.* We will first briefly discuss the functionally nonlinear type of state-space models in order to obtain a sense of arguably the more common meaning of nonlinear state-space models. Subsequently, we will elaborate on the conditionally Gaussian models. We note that in the subsequent chapters we will refer to the conditionally Gaussian models as just nonlinear, since the models that are explored in this research are either functionally linear and Gaussian models or functionally linear and conditionally Gaussian models.

Functionally Nonlinear State-Space Models

Then, having specified the different notions of nonlinearity, we return to the rainfall example. Recall that in this simple example we assumed that the actual rainfall at day t influences the actual rainfall at the previous day $t-1$ with some state-transition factor f and that the observation of the water level in the water tank is linked with the actual rainfall via some observation factor g as in (3.1). Nonlinearity in the state does not change the underlying dependence relations between the observations y_t and the states x_t that we discussed for linear state-space models. However, it changes the way our observations are connected with the states. In other words, we assume that the actual rainfall at day t does not influence the actual rainfall at day $t-1$ by some factor. Instead of a linear relationship, we could for example assume that there is some logistic relationship between the true rainfall at day t and day $t-1$. Additionally, for the sake of the example we could assume that the observation of the water level in the tank at day t is the square of the true rainfall at the same day t. Together, this results into the nonlinear Gaussian state-space model

$$y_t = g_t(x_t) + w_t = x_t^2 + w_t, \tag{3.9}$$

$$x_t = f_t(x_{t-1}) + v_t = \frac{1}{1+\exp\left(-x_{t-1}\right)} + v_t. \tag{3.10}$$

The logistic and square functions can be any nonlinear functions f_t and g_t, though. Furthermore, we can summarize this in the following definition based on Kitagawa (1996, p. 2) and Anderson and Moore (1979, p. 194).

Definition 3.2. *(Functionally Nonlinear Gaussian State-Space Model). A time-series model can be represented as a functionally nonlinear state-space model. By this we mean that the series* $\{\boldsymbol{y}_t, t = 0, 1, \dots\}$ *satisfies the equation*

$$\boldsymbol{y}_t = g_t(\boldsymbol{x}_t) + \boldsymbol{w}_t, \quad t = 0, 1, \dots, \tag{3.11}$$

$$\boldsymbol{x}_t = f_t(\boldsymbol{x}_{t-1}) + \boldsymbol{v}_t, \quad t = 0, 1, \dots, \tag{3.12}$$

where $f_t : \mathbb{R}^n \longrightarrow \mathbb{R}^n, g_t : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ *are nonlinear functions,* $\boldsymbol{y}_t \in \mathbb{R}^p, \boldsymbol{x}_t \in \mathbb{R}^n$ *for* $t = 0, 1, \dots$. *Moreover,* $\boldsymbol{w}_t$ *and* $\boldsymbol{v}_t$ *are independent and distributed as*

$$\begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{w}_t \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} R_t & S_t^T \\ S_t & Q_t \end{bmatrix}\right), \tag{3.13}$$

with $Q_t \in \mathbb{R}^{n \times n}, R_t \in \mathbb{R}^{p \times p}, S_t \in \mathbb{R}^{n \times p}$.

Conditionally Gaussian State-Space Models

A conditionally Gaussian state-space model is characterized by the fact that at least one of the system matrices is dependent of past observations. As a result the state-space model is Gaussian *given* that these past observations are known. A definition of a conditionally Gaussian state-space model is given in Definition 3.3 and is based on Harvey (1990, p. 156).

Definition 3.3. *(Conditionally Gaussian State-Space Model). A time-series model can be represented as a conditionally Gaussian state-space model. By this we mean that the series* $\{\boldsymbol{y}_t, t = 0, 1, \dots\}$ *satisfies the equation*

$$\boldsymbol{y}_t = G_t(Y_{t-1})\boldsymbol{x}_t + \boldsymbol{w}_t, \quad t = 0, 1, \dots, \tag{3.14}$$

$$\boldsymbol{x}_t = F_t(Y_{t-1})\boldsymbol{x}_{t-1} + \boldsymbol{v}_t, \quad t = 0, 1, \dots, \tag{3.15}$$

where $F_t \in \mathbb{R}^{n\times n}, G_t \in \mathbb{R}^{p\times n}, \boldsymbol{y}_t \in \mathbb{R}^p, \boldsymbol{x}_t \in \mathbb{R}^n$ *for* $t = 0, 1, \dots$ *and past observations are denoted by* $Y_{t-1} = \boldsymbol{y}_1, \dots, \boldsymbol{y}_{t-1}$*. Moreover,* $\boldsymbol{w}_t$ *and* $\boldsymbol{v}_t$ *are conditionally independent and distributed as*

$$\boldsymbol{w}_t|Y_{t-1} \sim \mathcal{N}(\boldsymbol{0}, R_t(Y_{t-1})), \tag{3.16}$$

$$\boldsymbol{v}_t|Y_{t-1} \sim \mathcal{N}(\boldsymbol{0}, Q_t(Y_{t-1})), \tag{3.17}$$

with $Q_t \in \mathbb{R}^{n\times n}$ *and* $R_t \in \mathbb{R}^{p\times p}$*.*

We note that in general the system matrices do not have to depend on all past observations until time $t-1$, as long as no dependency exists on a current or future observation. In Example 3.4 we elaborate on this type of state-space model.

Example 3.4. *Recall that the one-dimensional rainfall example in (3.1) is a linear state-space model defined as*

$$x_t = f x_{t-1} + v_t,\ v_t \sim \mathcal{N}(0, q^2), \tag{3.18}$$

$$y_t = g x_t + w_t,\ w_t \sim \mathcal{N}(0, r^2), \tag{3.19}$$

where x_t *is the true rainfall and* y_t *is the measurement of the rainfall. Now, it might be the case that the sensor measuring rainfall has worse accuracy during periods of high-intensity rainfall. Consequently, we could model the observation noise variance as*

$$r_t = b y_{t-1}, \tag{3.20}$$

which means that the variance is dependent on some fraction $b \in (0, 1)$ *of the measured rainfall of the past day* y_{t-1}*. Then, we can write the resulting state-space model as*

$$x_t \sim \mathcal{N}(f x_{t-1}, q^2), \tag{3.21}$$

$$y_t|y_{t-1} \sim \mathcal{N}(g x_t, b y_{t-1}), \tag{3.22}$$

from which it becomes clear that this state-space model is Gaussian given the past observation and, hence, is conditionally Gaussian.

Furthermore, we remark that Definition 3.3 can be generalized to also include past noise values $\boldsymbol{w}_1, \dots, \boldsymbol{w}_{t-1}$ and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{t-1}$ or past state values $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{t-1}$. As a result, for such models we need approximations of these values to estimate states with the Kalman filter, on which we elaborate in Section 3.3 and in more detail for the used type of yield curve model that incorporates volatility in Subsection 5.2.2.

3.2 Observability

In this section we discuss some of the key concepts for state-space models, namely observability and controllability. Intuitively, controllability says something about whether the model can be steered into some other state with control variables, whereas observability says something about whether the state variables can be explained by the observations. Using the multidimensional rainfall example, we can see the two concepts as follows. Controllability says something about whether we can use, for instance, irrigation in such a way that the final state (like groundwater level) is as desired. Then, observability says something about whether the true rainfall, groundwater levels and soil moisture can be directly observed with the used measurements. The two properties, controllability and observability, are dual properties as Kalman (1960, p. 499) describes and the way these two properties are related is given in Definition 3.5. The duality means that observability implies controllability and vice versa. For a more thorough derivation we refer to the original paper (Kalman, 1960, pp. 498-499).

Both properties are generally important to consider for state-space models. However, in this research observability is a more relevant notion as we do not use additional external variables. Specifically, if a state-space model is *observable*, it means that the unobservable state variable can exclusively be determined from the observations. Because of this, we will only elaborate on observability in the remainder of the section. We define observability in Definition 3.5, which is given in Brockwell and Davis (1991, p. 496) as Proposition 12.4.4.

Definition 3.5. *(Observability). The pair of matrices (F, G) is observable if and only if O_n has rank n. In particular, (F, G) is observable if and only if (F^T, G^T) is controllable. Here, the matrix $O_k \in \mathbb{R}^{kp \times n}$ for some $k \in \mathbb{N} \cup \{0\}$ is defined as*

$$O_k = \begin{bmatrix} G \\ GF \\ \vdots \\ GF^{k-1} \end{bmatrix}, \tag{3.23}$$

and is called the observability matrix.

Notice that in Definition 3.5 the components of O_k only go until the $k-1$-th power of F due to the so-called *Cayley-Hamilton Theorem*. It goes beyond the aim of this section to go into full-depth of this theorem, but the key idea is that a for a square matrix $A \in \mathbb{R}^{n \times n}$, the n-th power A^n can be written as a linear combination of $A, A^2, \ldots, A^{n-1}$. For the details we refer to Brockwell and Davis (1991, p. 492).

Showing observability of a state-space model by using Definition 3.5 can be a tedious task as the observability matrix O_k can become large. The results that are shown below provide some sufficient conditions to show that a state-space model is observable and are more straightforward. Lemma 3.7 is equivalent to Lemma 3.6 (*Hautus Lemma* or sometimes also called *Popov-Belevitch-Hautus (PBH)* test) and, in practice, is a useful result in showing whether a state-space model is observable. We do not provide the proof of Lemma 3.6, but we will prove the equivalence of Lemma 3.7 with Lemma 3.6.

Lemma 3.6. *(Lemma 3.3.7 (Hautus lemma), Sontag (1998, p. 272)). Let $F \in \mathbb{R}^{n \times n}$ and $G \in \mathbb{R}^{p \times n}$. The following properties are equivalent for the pair (F, G).*

1. (F, G) is observable.

2. $\operatorname{rank}\left(\begin{bmatrix} \lambda I - F \\ G \end{bmatrix}\right) = n$ *for all* $\lambda \in \mathbb{C}$.

3. $\operatorname{rank}\left(\begin{bmatrix} \lambda I - F \\ G \end{bmatrix}\right) = n$ *for each eigenvalue* λ *of* F.

Proof. For the proof, see Sontag (1998), pp. 94-95. □

Lemma 3.7 reduces the task of testing for observability of a linear state-space model to only computing the eigenvalues of F and confirming whether the associated eigenvectors multiplied with observation matrix G result into the zero vector $\mathbf{0}$. The lemma is as follows.

Lemma 3.7. *The pair (F, G) is observable if and only if there exists no $\boldsymbol{v} \neq \mathbf{0}$ such that $F\boldsymbol{v} = \lambda\boldsymbol{v}$ and $G\boldsymbol{v} = \mathbf{0}$.*

Proof. We will prove Lemma 3.7 by proving equivalence between the contraposition of the third point of Lemma 3.6 and Lemma 3.7. First, we write

$$A := \begin{bmatrix} \lambda I - F \\ G \end{bmatrix}. \tag{3.24}$$

Assume that the contraposition of the third point of Lemma 3.6 holds, which is

$$\operatorname{rank}(A) < n, \tag{3.25}$$

for each eigenvalue λ of F. Then, the columns of A are linearly dependent. From this it follows that there exists a $\boldsymbol{v} \in \mathbb{R}^n$ such that $A\boldsymbol{v} = \mathbf{0}$. Now, it directly follows from writing out A that

$$A\boldsymbol{v} = \begin{bmatrix} \lambda I - F \\ G \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} (\lambda I - F)\boldsymbol{v} \\ G\boldsymbol{v} \end{bmatrix} = \mathbf{0}, \tag{3.26}$$

$$\Leftrightarrow$$

$$F\boldsymbol{v} = \lambda\boldsymbol{v} \text{ and } G\boldsymbol{v} = \mathbf{0}. \tag{3.27}$$

From Lemma 3.6 we know that this is equivalent with the pair (F, G) being unobservable, which proves Lemma 3.7. □

For a diagonal matrix F the eigenvalue and eigenvector computations become trivial. Since we only use diagonal state-transition matrices in the used state-space models, this specific case is summarized in Corollary 3.8.

Corollary 3.8. *If $F \in \mathbb{R}^{n \times n}$ is a diagonal matrix and $G \in \mathbb{R}^{p \times n}$ has only nonzero columns, then the pair (F, G) is observable.*

Proof. Define $F \in \mathbb{R}^{n \times n}$ and $G \in \mathbb{R}^{p \times n}$ as

$$F = \begin{bmatrix} f_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & f_n \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & \dots & G_n \end{bmatrix}, \tag{3.28}$$

where G_i denotes the i-th column of matrix G. Since F is a diagonal matrix, it follows immediately that the eigenvalues of F are $f_1, \dots, f_n$. Consequently, the eigenvector $\boldsymbol{v}_i$ corresponding to the eigenvalue f_i is the standard unit vector, i.e. $\boldsymbol{v}_i = \boldsymbol{e}_i$ for $i = 1, \dots, n$. Since matrix G has only nonzero columns, i.e. $G_i \neq \mathbf{0}$ for $i = 1, \dots, n$, it holds for every eigenvector $\boldsymbol{v}_i = \boldsymbol{e}_i$ that $G\boldsymbol{e}_i = G_i \neq \mathbf{0}$. So, there exists no $\boldsymbol{v} \neq \mathbf{0}$ such that $F\boldsymbol{v} = \lambda\boldsymbol{v}$ and $G\boldsymbol{v} = \mathbf{0}$. Hence, by Proposition 3.7, we conclude that the pair (F, G) is observable. □

If we want to modify the original Dynamic Nelson-Siegel model, Corollary 3.8 can be used to check that the extensions are indeed observable. This result ensures us that the state estimates obtained through the Kalman filter are optimal and that the observations give enough information for estimating the states, on which we will elaborate in the next section.

Example 3.9. *We consider two models in this example.*

1. *For the Dynamic Nelson-Siegel (DNS) model we have to consider the pair* (Φ, Λ)*. As* $\Phi = \operatorname{diag}(\phi_1, \phi_2, \phi_3)$ *and* Λ *has no columns with only zeros, it follows directly from Corollary 3.8 that the DNS model is observable.*

2. *For the DNS model with observation noise following an autoregressive process of order 1 and states modeled by a random walk (DNS-ARRW) we have to consider the pair* $(\tilde{\Phi}, \tilde{\Lambda})$*. In this case* $\tilde{\Phi} = \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix}$*, where* I *and* A *are diagonal matrices with diagonals* $(1, 1, 1, \alpha_1, \dots, \alpha_{11})$*.*

   For the observation matrix we have $\tilde{\Lambda} = \begin{bmatrix} \Lambda & I \end{bmatrix}$*, so* $\tilde{\Lambda}$ *has only nonzero columns. Hence, from Corollary 3.8 it follows that* $(\tilde{\Phi}, \tilde{\Lambda})$ *is observable.*

3.3 State Estimation

In this section we discuss one of the most important concepts in regard to state-space models, *state estimation* (Shumway and Stoffer, 2011; Kitagawa, 1996). In the context of state-space models, estimating the state variable $\boldsymbol{x}_t$ means finding some estimate of $\boldsymbol{x}_t$ given observations $Y_n = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$. Recall the example where the state is given by the true rainfall, which we try to measure with a water tank. Every day we observe the water level in the water tank, which gives us a collection of observations $Y_t = \{y_1, y_2, \dots, y_t\}$. However, we are actually interested in the true rainfall every day $x_1, x_2, \dots, x_t$. So, we need some way to infer x_t from observations Y_t. In other words, for each day t we would like to know the true rainfall given that we know the water levels in the tank from day 1 to day t, which is summarized by $\mathbb{E}[x_t|Y_t]$. Notice that since the true rainfall and the observations have some randomness, $\mathbb{E}[x_t|Y_t]$ has some randomness as well. That is why we are also interested in the "accuracy" of an estimate. This is encapsulated in the variance of the true rainfall x_t given our observations Y_t, denoted by $\mathrm{Var}[x_t|Y_t]$. In short, we are interested in the *filter density* $p(x_t|Y_t)$.

In the proceeding parts of this section we present a method to estimate the states $x_1, \dots, x_t$ by using the filter density called the Kalman filter originally derived by Kalman (1960). Particularly, we only focus on the Kalman filter and some important notions related to this filter, since it can be used for linear Gaussian state-space models and with some approximations for conditionally Gaussian state-space models as well. In order to make the following parts of this section more readable, we first introduce some notation that Shumway and Stoffer (2011) use as well.

Notation.

$$\boldsymbol{x}_t^s := \mathbb{E}[\boldsymbol{x}_t|Y_s], \tag{3.29}$$

$$P_{t_1,t_2}^s := \mathbb{E}[(\boldsymbol{x}_{t_1} - \boldsymbol{x}_{t_1}^s)(\boldsymbol{x}_{t_2} - \boldsymbol{x}_{t_2}^s)^T], \tag{3.30}$$

where $Y_s = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_s\}$*. When* $t_1 = t_2 = t$*, we obtain*

$$P_t^s := \mathbb{E}[(\boldsymbol{x}_t - \boldsymbol{x}_t^s)^2]. \tag{3.31}$$

Moreover, we assume that the processes $\boldsymbol{x}_t, \boldsymbol{y}_t$ are normally distributed. Shumway and Stoffer (2011) show that $(\boldsymbol{x}_t - \boldsymbol{x}_t^s) \perp Y_s$ for $s, t > 0$, from which it follows that

$$\begin{aligned} P_t^s &= \mathbb{E}[(\boldsymbol{x}_t - \boldsymbol{x}_t^s)^2], && (3.32)\\ &= \mathbb{E}[(\boldsymbol{x}_t - \boldsymbol{x}_t^s)^2|Y_s] \quad \text{(normality)} && (3.33)\\ &= \mathbb{E}[(\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_t|Y_s])^2|Y_s] \quad \text{(definition 3.3)}, && (3.34)\\ &=: \mathrm{Var}[\boldsymbol{x}_t|Y_s] \quad \text{(definition Var}(\cdot)). && (3.35) \end{aligned}$$

3.3.1 Kalman Filter for Linear Gaussian SSMs

The Kalman filter is based on a two-step approach, consisting of the *prediction* step and the *filtering* step. We introduce the Kalman filter through the one-dimensional rainfall example with state-space model (3.1). First we give an outline of the basic idea before discussing the derivation of the Kalman filter later in this subsection. The first step is to *predict* the state x_t using observations until time $t-1$. Since the observations Y_{t-1} and the way x_t is defined are both known, we can derive that

$$\begin{aligned} x_t^{t-1} &= \mathbb{E}[x_t|Y_{t-1}] = \mathbb{E}[fx_{t-1} + v_t|Y_{t-1}] = fx_{t-1}^{t-1}, && (3.36)\\ P_t^{t-1} &= \mathrm{Var}[x_t|Y_{t-1}] = \mathrm{Var}[fx_{t-1} + v_t|Y_{t-1}] = f^2 P_{t-1}^{t-1} + q^2. && (3.37) \end{aligned}$$

Then, the idea of the filtering step is to improve the predicted state x_t^{t-1} with the new information that observation y_t gives. However, the new information that y_t gives is relative to the predicted observation gx_t^{t-1}, which says something about how close the model for the state predicts the observations compared with the actual observation. That is why we consider the residual $\epsilon_t = y_t - gx_t^{t-1}$ instead of the observation y_t, which is called the innovation term. The predicted state x_t^{t-1} is improved by finding some optimal combination of the predicted state and the new information that y_t gives. Another way to put it is by writing

$$x_t^t = k_t y_t + (1 - k_t) g x_t^{t-1} = g x_t^{t-1} + k_t(y_t - g x_t^{t-1}) = g x_t^{t-1} + k_t \epsilon_t, \tag{3.38}$$

where k_t is called the Kalman gain, which tells us how much the new observation should "weigh" in the improved estimate x_t^t. How much new information the observation y_t gives, is intuitively connected with how accurate an observation is (Bar-Shalom et al., 2001, p. 207). Recall that the observation noise is $w_t \sim \mathcal{N}(0, r^2)$. Therefore, a large r means that the new observation y_t gives weaker information than if r is small and, consequently, the observation is accurate. The Kalman gain k_t actually depends partly on the observation noise r and is the weight that minimizes $P_t^t = \text{Var}[x_t^t]$, the uncertainty of x_t^t.

In summary, there are four essential variables to be considered when estimating states. That is,

- $\boldsymbol{y}_t$: the original observation;
- $\boldsymbol{x}_t$: the *true* state;
- $\boldsymbol{x}_t^t$: the state estimated at time t given observation till time t;
- P_t^t: the uncertainty of the estimated state at time t given observations till time t;

Bishop (2006. p. 641) illustrates these variables with a two-dimensional example, which serves as a visualisation of the Kalman filter. In Figure 3.2 the original illustration is shown. The most important part of this illustration is the fact that we cannot say that the estimated states are the *true* states with 100% certainty. This uncertainty is captured with the red circles representing P_t^t around the red crosses representing $\boldsymbol{x}_t^t$.

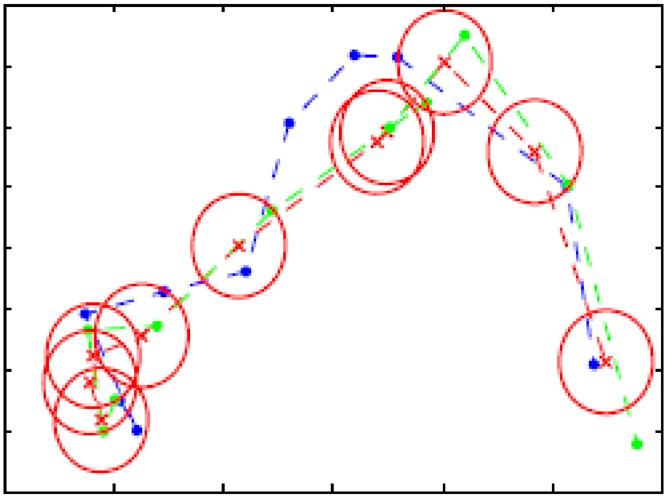


Figure 3.2: This is an example of a state-space model used to track movements. The blue points are the states (true position) $\boldsymbol{x}_t$, the green points are the measurements $\boldsymbol{y}_t$ and the red crosses are the estimated states $\boldsymbol{x}_t^t$ obtained by the Kalman filter. The red circles around the state estimates represent the covariance P_t^t of each state estimate. (*Source*: Bishop, 2006, p. 641).

The described concepts of state estimation can be formalized for a state-space model as in Definition 3.1 in the following theorem, which Shumway and Stoffer (2011) provide as Property 6.1 on p. 326.

Theorem 3.10. *(Kalman Filter) For the state-space model specified as in Definition 3.1 with initial conditions $\boldsymbol{x}_0^0 = \boldsymbol{\mu}_0$ and $P_0^0 = \Sigma_0$, for $t = 1, \dots, T$ we have*

$$\begin{aligned} \boldsymbol{x}_t^{t-1} &= F\boldsymbol{x}_{t-1}^{t-1}, && (3.39)\\ P_t^{t-1} &= FP_{t-1}^{t-1}F^T + Q, && (3.40)\\ & && (3.41) \end{aligned}$$

with

$$\begin{aligned} \boldsymbol{x}_t^t &= \boldsymbol{x}_t^{t-1} + K_t(\boldsymbol{y}_t - G\boldsymbol{x}_t^{t-1}), && (3.42)\\ P_t^t &= [I - K_tG]P_t^{t-1}, && (3.43) \end{aligned}$$

where

$$K_t = P_t^{t-1}G^T[GP_t^{t-1}G^T + R]^{-1} \qquad (3.44)$$

is called the Kalman gain. Prediction for $t > T$ can be done with initial conditions $\boldsymbol{x}_T^T$ and P_T^T. Moreover, we call

$$\begin{aligned} \boldsymbol{\epsilon}_t &= \boldsymbol{y}_t - \boldsymbol{y}_t^{t-1} = \boldsymbol{y}_t - \mathbb{E}[\boldsymbol{y}_t|Y_{t-1}] && (3.45)\\ &= \boldsymbol{y}_t - G\boldsymbol{x}_t^{t-1}, && (3.46) \end{aligned}$$

innovations/prediction errors and define the corresponding variance-covariance matrices

$$\begin{aligned} \Sigma_t &:= \mathrm{Var}[\boldsymbol{\epsilon}_t] = \mathrm{Var}[G(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}) + \boldsymbol{w}_t] && (3.47)\\ &= GP_t^{t-1}G^T + R, && (3.48) \end{aligned}$$

for $t = 1, \dots, T$.

Although a formal proof of the Kalman filter is provided by Shumway and Stoffer (2011, pp. 326-327), it goes somewhat quickly through important steps. That is why we will elaborate more on the outline of the proof. We skip the proof of the prediction step, because it is a straightforward computation similar to the computation for the one-dimensional example. The proof of the filtering step begins with writing out $\boldsymbol{x}_t^t$ and using that $\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - G\boldsymbol{x}_t^{t-1}$ and $\boldsymbol{y}_s$ are independent for $s < t$

$$\boldsymbol{x}_t^t = \mathbb{E}[\boldsymbol{x}_t|Y_t] = \mathbb{E}[\boldsymbol{x}_t|Y_{t-1}, \boldsymbol{\epsilon}_t]. \qquad (3.49)$$

In order to derive the conditional expectation Shumway and Stoffer (2011) use a lemma that is key in their proof. The lemma states that for two random variables $X = (X_1, \dots, X_n), Y = (Y_1, \dots, Y_m)$, if their joint distribution is Gaussian with

$$\begin{bmatrix} Y \\ X \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_Y \\ \mu_X \end{bmatrix}, \begin{bmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{bmatrix} \right), \qquad (3.50)$$

then the conditional distribution is also Gaussian. In other words, $Y|X \sim \mathcal{N}(\mu_{Y|X}, \Sigma_{Y|X})$ with $\mu_{Y|X}$ and $\Sigma_{Y|X}$ given by

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(X - \mu_X), \tag{3.51}$$

$$\Sigma_{Y|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}. \tag{3.52}$$

We will not prove this lemma, but it can be shown by working out the definition of the conditional $f_{Y|X}(y|x) = \frac{f_{Y,X}(y,x)}{f_X(x)}$. Using this lemma, the proof reduces to showing that $\boldsymbol{\epsilon}_t$ is Gaussian and computing its mean and variance, and the covariance between $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_t$. The authors use that $\boldsymbol{\epsilon}_t \perp Y_s$, and show that $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_t)$. Moreover, the covariance is then also a quite straightforward computation and given by $\mathrm{Cov}[\boldsymbol{x}_t, \boldsymbol{\epsilon}_t|Y_{t-1}] = P_t^{t-1}G^T$, of which the computation is already given by the authors. Since we have all components of the joint distribution $\boldsymbol{x}_t, \boldsymbol{\epsilon}_t|Y_{t-1}$, this means all the components are known for the conditional distribution $\boldsymbol{x}_t|Y_{t-1}, \boldsymbol{\epsilon}_t$

$$\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\epsilon}_t \end{bmatrix} \Bigg| Y_{t-1} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}_t^{t-1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} P_t^{t-1} & P_t^{t-1}G^T \\ GP_t^{t-1} & \Sigma_t \end{bmatrix} \right). \tag{3.53}$$

From the lemma $\boldsymbol{x}_t^t$ and P_t^t follow directly

$$\begin{aligned} \boldsymbol{x}_t^t = \mathbb{E}[\boldsymbol{x}_t|Y_t] &= \mathbb{E}[\boldsymbol{x}_t|Y_{t-1}, \boldsymbol{\epsilon}_t] && (3.54)\\ &= \boldsymbol{x}_t^{t-1} + P_t^{t-1}G^T\Sigma_t^{-1}(\boldsymbol{\epsilon}_t) && (3.55)\\ &= x_t^{t-1} + K_t\boldsymbol{\epsilon}_t, && (3.56)\\ P_t^t = \mathrm{Var}[\boldsymbol{x}_t|Y_t] &= \mathrm{Var}[\boldsymbol{x}_t|Y_{t-1}, \boldsymbol{\epsilon}_t] && (3.57)\\ &= P_t^{t-1} - P_t^{t-1}G^T\Sigma_t^{-1}GP_t^{t-1} && (3.58)\\ &= P_t^{t-1} - K_tGP_t^{t-1} && (3.59)\\ &= (I - K_tG)P_t^{t-1}. && (3.60) \end{aligned}$$

where $K_t := P_t^{t-1}G^T\Sigma_t^{-1} = P_t^{t-1}G^T\left(GP_t^{t-1}G^T + R\right)^{-1}$ is the Kalman gain. As mentioned for the one-dimensional example, we can see that the filtered state estimate is indeed some weighted combination of the predicted state estimate $\boldsymbol{x}_t^{t-1}$ and the innovation $\boldsymbol{\epsilon}_t$. Here, the Kalman gain K_t can be seen as a weight again that indicates to what extent the state estimate should rely on the prediction step $\boldsymbol{x}_t^{t-1}$ or the new observation $\boldsymbol{y}_t$ partly depending on the observation noise R. If the observations are not very accurate, i.e. R has large entries, then one can imagine that this will contribute to Σ_t^{-1} (and thus K_t) becoming small. Consequently, a small K_t will give “less weight” to the innovation term resulting into $\boldsymbol{x}_t^t$ being much more close to the predicted $\boldsymbol{x}_t^{t-1}$.

Notice that the Kalman filter also needs two initial conditions. Before we can employ the Kalman filter, we have to give an initial state $\boldsymbol{x}_0^0$ and an initial covariance matrix P_0^0. In other words we need to have some idea how the first state is distributed, where $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{x}_0^0, P_0^0)$. In general, we can consider these initial conditions to be unknown parameters like the parameters in G, R, F and Q resulting into a collection of parameters $\boldsymbol{\theta} := \{\boldsymbol{x}_0^0, P_0^0, G, R, F, Q\}$.

Numerical Stability

The Kalman filter as presented in Theorem 3.10 computes P_t^t as

$$P_t^t = (I - K_t G)P_t^{t-1} = P_t^{t-1} - K_t G P_t^{t-1}, \tag{3.61}$$

where the implicit assumption is that the Kalman gain K_t is optimal in the sense that it minimizes the covariance matrix P_t^t. However, in practice, there may be situations where the entries of the noise matrices R, Q are so small that numerical precision is at stake. In these cases, the Kalman gain K_t is not optimal anymore, which can result into state estimates that are not optimal anymore. In order to counter numerical instability we use the so-called *Joseph form* of the computation of P_t^t defined as

$$P_t^t = (I - K_t G)P_t^{t-1}(I - K_t G)^T + K_t R K_t^T. \tag{3.62}$$

The Joseph form requires more computational effort, which can make the filter using (3.62) slower compared to using (3.61), but less prone to round-off errors (Bar-Shalom et al., 2001, p. 206). Expression (3.61) is actually a simplified variant of the Joseph form, which is the result of the fact that $K_t = P_t^{t-1} G^T \Sigma_t^{-1}$ is the Kalman gain minimizing P_t^t. The derivation is based on writing out the definition of P_t^t as in (3.32) and then plugging in all the necessary definitions (Problem 5-5 in Bar-Shalom et al., 2001, p. 262).

$$\begin{aligned}
P_t^t &= \mathrm{Cov}(\boldsymbol{x}_t - \boldsymbol{x}_t^t) &\quad (3.63)\\
&= \mathrm{Cov}(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1} - K_t(\boldsymbol{y}_t - G\boldsymbol{x}_t^{t-1})) &\quad (3.64)\\
&= \mathrm{Cov}(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1} - K_t(G\boldsymbol{x}_t + \boldsymbol{w}_t - G\boldsymbol{x}_t^{t-1})) &\quad (3.65)\\
&= \mathrm{Cov}((I - K_t G)(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1})) + \mathrm{Cov}(K_t \boldsymbol{w}_t) &\quad (3.66)\\
&= (I - K_t G)P_t^{t-1}(I - K_t G)^T + K_t R K_t^T. &\quad (3.67)
\end{aligned}$$

Then, we know that the optimal Kalman gain is given by $K_t = P_t^{t-1} G^T \Sigma_t^{-1}$, which is the same as $K_t \Sigma_t = P_t^{t-1} G^T$. So, the simplified form (3.61) follows directly after rewriting the Joseph form and substituting the optimal $K_t \Sigma_t$

$$\begin{aligned}
P_t^t &= (I - K_t G)P_t^{t-1}(I - K_t G)^T + K_t R K_t^T &\quad (3.68)\\
&= P_t^{t-1} - K_t G P_t^{t-1} - P_t^{t-1} G^T K_t^T + K_t \Sigma_t K_t^T \text{ (definition } \Sigma_t) &\quad (3.69)\\
&= P_t^{t-1} - K_t G P_t^{t-1} - P_t^{t-1} G^T K_t^T + P_t^{t-1} G^T K_t^T \text{ (definition optimal } K_t) &\quad (3.70)\\
&= P_t^{t-1} - K_t G P_t^{t-1}. &\quad (3.71)
\end{aligned}$$

Log-likelihood

Previously, we defined the collection of all unknown parameters of a state-space model and initial conditions for the Kalman filter as $\boldsymbol{\theta} := \{\boldsymbol{x}_0^0, P_0^0, G, R, F, Q\}$. Then, an important advantage of using the Kalman filter to estimate the states is that by computing $\boldsymbol{\epsilon}_t$ and Σ_t for every iteration we also have the necessary ingredients for computing the likelihood $p(Y_t|\boldsymbol{\theta})$, and thus the log-likelihood $\ell(\boldsymbol{\theta}; Y_t) := \log(p(Y_t|\boldsymbol{\theta}))$. This is a direct result of the fact that the Kalman filter works on linear Gaussian state-space models. The linearity of the model ensures that the Gaussian terms stay Gaussian, while the Gaussian noise terms let us derive a closed-form expression for the log-likelihood. Durbin and Koopman (2012, p. 171) give a brief derivation, which we will elaborate on.

Suppose we have n observations $Y_n = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)$ and $\boldsymbol{y}_t \in \mathbb{R}^p$ for $t = 1, \dots, n$, then we can write the likelihood as a product of conditional densities using the definition of a conditional density.

$$\begin{aligned} p(Y_t|\boldsymbol{\theta}) := p(Y_t) &= p(Y_{t-1})p(\boldsymbol{y}_t|Y_{t-1}) && (3.72)\\ &= p(Y_{t-2})p(\boldsymbol{y}_{t-1}|Y_{t-2})p(\boldsymbol{y}_t|Y_{t-1}) && (3.73)\\ &\vdots \\ &= p(\boldsymbol{y}_1)\prod_{t=2}^{n} p(\boldsymbol{y}_t|Y_{t-1}). && (3.74) \end{aligned}$$

Deriving the likelihood now consists of computing every density $p(\boldsymbol{y}_t|Y_{t-1})$. In particular, we have

$$\begin{aligned} \mathbb{E}[\boldsymbol{y}_t|Y_{t-1}] &= \mathbb{E}[G\boldsymbol{x}_t + \boldsymbol{w}_t|Y_{t-1}] && (3.75)\\ &= G\mathbb{E}[\boldsymbol{x}_t|Y_{t-1}] && (3.76)\\ &= G\boldsymbol{x}_t^{t-1}, && (3.77)\\ \mathrm{Var}[\boldsymbol{y}_t|Y_{t-1}] &= \mathrm{Var}[G\boldsymbol{x}_t + \boldsymbol{w}_t|Y_{t-1}] && (3.78)\\ &= G\mathrm{Var}[\boldsymbol{x}_t|Y_{t-1}]G^T + R && (3.79)\\ &= GP_t^{t-1}G^T + R. && (3.80) \end{aligned}$$

We deduce that $p(\boldsymbol{y}_t|Y_{t-1}) \overset{d}{=} \mathcal{N}\left(G\boldsymbol{x}_t^{t-1}, GP_t^{t-1}G^T + R\right)$, which can be rewritten in a more compact form in terms of the innovation $\boldsymbol{\epsilon}_t$ by using that $G\boldsymbol{x}_t^{t-1} = \boldsymbol{y}_t - \boldsymbol{\epsilon}_t$ and $\Sigma_t = GP_t^{t-1}G^T + R$. Hence, $p(\boldsymbol{y}_t|Y_{t-1}) \overset{d}{=} \mathcal{N}\left(\boldsymbol{y}_t - \boldsymbol{\epsilon}_t, \Sigma_t\right)$ and the log-likelihood $\ell(Y_t|\boldsymbol{\theta})$ is given by

$$\begin{aligned} \ell(Y_t|\boldsymbol{\theta}) &= \log\left\{p(\boldsymbol{y}_1)\prod_{t=2}^{n} p(\boldsymbol{y}_t|Y_{t-1})\right\} && (3.81)\\ &= -\frac{np}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\log|\Sigma_t| - \frac{1}{2}\sum_{t=1}^{n}\boldsymbol{\epsilon}_t^T\Sigma_t^{-1}\boldsymbol{\epsilon}_t. && (3.82) \end{aligned}$$

The log-likelihood formulated in terms of the innovation $\boldsymbol{\epsilon}_t$ and its variance Σ_t as in (3.82) will be useful for comparing different models and for estimating parameters with the Markov Chain Monte Carlo method that we use, as we will discuss in Chapter 4.

3.3.2 Modified Kalman Filter for Conditionally Gaussian SSMs

Recall that a more general conditionally Gaussian state-space model can consist of system matrices G_t, F_t, R_t, Q_t that are not only dependent on past observations Y_{t-1}, but also on past values of the state $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{t-1}$ or the noise terms $\boldsymbol{w}_1, \dots, \boldsymbol{w}_{t-1}$ and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{t-1}$. If such a model has dependence on past observations, the resulting state-space model has time-variant system matrices, for which we can still use the Kalman filter. On the other hand, if such a model has dependence on past states or noise terms, then the Kalman filter as in Theorem 3.10 needs approximations for these values as they are assumed to be not directly observable. However, there is no general approximation that works for all types of conditionally Gaussian state-space models while still having the advantages of a Kalman filter. So, the approximations depend on what kind of conditionally Gaussian state-space model is considered. This is in contrast with the functionally nonlinear state-space models, for which the extended[1] or unscented[2] Kalman filter is a general method especially developed for this kind of nonlinearity.

Subsequently, we do not provide all ways in which the Kalman filter can be modified for a conditionally Gaussian state-space model. In particular, we refer to Subsection 5.2.2, in which we elaborate on the approximation that is used for the specific volatility model that has conditionality on the observation noise term.

[1] See Anderson and Moore (1979, p. 195)

[2] See Wan and Van der Merwe (2000)

4

Bayesian Parameter Estimation and Forecasting

In this chapter we will discuss the way we estimate parameters. We begin this chapter by giving a brief introduction on Bayesian statistics in Section 4.1. In this section we explain some important concepts such as the prior and posterior distributions to estimate parameters. In the same section we will also elaborate on forecasting in a Bayesian setting with the posterior predictive distribution. Afterwards, we discuss Markov Chain Monte Carlo (MCMC) methods and specifically the Random Walk Metropolis algorithm in Section 4.2, which is a method to approximate probability distributions (such as the posterior distribution).

4.1 A Brief Introduction to Bayesian Statistics

Suppose we have some data $\boldsymbol{x} = (x_1, \ldots, x_n)$ that is a realization of a random variable $X = (X_1, \ldots, X_n)$ with a distribution $p(\boldsymbol{x}|\theta)$. Then $\theta \in D \subset \mathbb{R}$ is the parameter of the distribution p and D denotes the support, which are the values that θ can attain. Usually, one is interested in estimating the parameter in order to perform analyses that are based on the distribution with the estimated parameter, denoted by $p(\boldsymbol{x}; \hat{\theta})$. Then, in classical statistics, or also the frequentist approach, the parameter θ is regarded as a fixed and unknown point (Young and Smith, 2005, p. 22). A popular method in finding the parameter estimator $\hat{\theta}$ is the maximum likelihood estimation method (Robert and Casella, 2004, p. 6). The frequentist approach consists of computing the likelihood function, denoted by $L(\theta|\boldsymbol{x})$ and defined as

$$\begin{aligned} L(\theta|\boldsymbol{x}) &= p(x_1, \ldots, x_n|\theta) && (4.1) \\ &= \prod_{i=1}^{n} p(x_i|\theta), \text{ (if } x_i\text{'s i.i.d.)}, && (4.2) \end{aligned}$$

where i.i.d. means independent and identically distributed. Subsequently, a frequentist will use the likelihood function to find the maximum likelihood estimator $\hat{\theta}$, that maximizes $L(\theta|\boldsymbol{x})$. The estimator $\hat{\theta} = \arg\max_{\theta \in D} L(\theta|\boldsymbol{x})$ is in this context a point estimate of θ. In contrast, in the Bayesian approach it is assumed that θ is a random variable like X. A Bayesian statistician has some belief about parameter θ before seeing the data x based on some prior information. This prior information is encapsulated in the *prior distribution* $p(\theta)$, or prior in short. Then, the likelihood, which in the Bayesian setting is often denoted by $p(\boldsymbol{x}|\theta)$, can be seen as the information that the observations $x_1, \dots, x_n$ give us. Together, the likelihood and the prior are related to the *posterior distribution* or posterior in short, denoted by $p(\theta|\boldsymbol{x})$, in the following way.

$$p(\theta|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\theta)p(\theta)}{p(\boldsymbol{x})} \tag{4.3}$$

$$= \frac{p(\boldsymbol{x}|\theta)p(\theta)}{\int p(\boldsymbol{x}|\theta')p(\theta')d\theta'} \tag{4.4}$$

$$\propto p(\boldsymbol{x}|\theta)p(\theta), \tag{4.5}$$

where we used the definition of a conditional distribution and a marginal distribution to express the posterior in terms of the likelihood and the prior. The term $\int p(\boldsymbol{x}|\theta')p(\theta')d\theta'$ is called the normalizing constant and ensures that the posterior integrates to one over the entire support D,

$$\int_{\theta' \in D} p(\theta'|\boldsymbol{x})d\theta' = 1, \tag{4.6}$$

and ensures that the posterior satisfies the definition of a probability distribution. However, for parameter estimation the normalizing constant is often ignored as it suffices to know the posterior up to a constant. Robert and Casella (2004, pp. 51-52) discuss this in further detail.

4.1.1 Choice of Prior and Posterior Derivation

As mentioned, the prior distribution can be considered as our belief on the parameter *prior* to seeing any data. When choosing priors, a question that arises is whether we want a prior to be *informative* or *non-informative*. An informative prior is one that gives some information about the parameter based on, for instance, historical data or some subjective grounds. One can imagine that the less data is available, the more the posterior will resemble the prior. In contrast, an non-informative prior is one that gives as less as possible information about the parameter. As a result, the posterior will depend mostly on the data. Young and Smith (2005) give four approaches for choosing a prior, of which we give the three approaches that are most common.

- *flat or uniform priors* are priors that are generally considered as non-informative priors. Examples include uniform priors such as $\theta \sim \mathcal{U}(0, 1)$, improper priors like $p(\theta) \propto 1$, or the so-called *Jeffreys prior*, which is a type of prior that is invariant under reparametrization. Invariance under reparametrization means that it does not matter whether, for instance, σ or σ^2 is used as a parameter for the variance.

- *Subjective priors* are priors that involve some expert judgement to assess prior beliefs or beliefs based on historical data.

- *Convenient priors* are priors that are mainly used to simplify the derivation of the posterior. Conjugate priors are a good example of such priors. These kinds of prior assure that the posterior and the prior are in the same probability density family.

Additionally, flat priors and Jeffreys prior are so-called *improper priors*. Improper priors are priors that do not satisfy condition (4.6), so $\int_{\theta' \in D} p(\theta')d\theta' \neq 1$. In general, an improper prior is acceptable as long as it does not result in an improper posterior. This is the case as long as $\int_{\theta' \in D} p(\theta'|x)d\theta' < +\infty$. As a consequence, if the likelihood multiplied with the prior resembles some known probability density up to a constant, then the posterior is proper. Gelman et al. (2014, p. 52 and Chapter 4) elaborate more on this. We give a small example, where we have one normally distributed observation and where we choose an improper prior.

Example 4.1. *(Improper and non-informative prior). A Debt Management Office (DMO) wants to quantify its knowledge about yields given some observation. Suppose that they have one observation of a 10 year government bond yield, which is $x = 1\%$. The DMO assumes that the yields are distributed according to a normal distribution with unknown mean μ and known variance $\sigma^2 = 1$, so $x \sim \mathcal{N}(\mu, 1)$. Now, the DMO needs to specify a prior for parameter μ, but does not know a lot about the mean. The only thing the DMO is sure of is that $\mu > 0$ (perhaps the DMO lives pre-2008). So, an improper prior is chosen as $p(\mu) \propto \mathbf{1}_{(0,+\infty)}(\mu)$, where $\mathbf{1}_A(\cdot)$ is the indicator function on some set A. Then, we will derive the posterior distribution for this example, which is given by*

$$p(\mu|x) = \frac{p(x|\mu)p(\mu)}{\int p(x|\mu')p(\mu')d\mu'}. \tag{4.7}$$

We compute the numerator and denominator separately. The derivation of the numerator is straightforward.

$$p(x|\mu)p(\mu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu)^2} \mathbf{1}_{(0,+\infty)}(\mu), \tag{4.8}$$

where $\mathbf{1}(\cdot)$ is the indicator function, which has value 1 if $\mu \in (0, +\infty)$ and 0 elsewhere. Subsequently, the denominator can be computed as follows.

$$\begin{aligned} \int p(x|\mu')p(\mu')d\mu' &= \int \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu')^2} \mathbf{1}_{(0,+\infty)}(\mu')d\mu' && (4.9) \\ &= \int_0^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu')^2} d\mu' && (4.10) \\ &= \int_0^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mu'-x)^2} d\mu'. && (4.11) \end{aligned}$$

Notice that we can recognize the probability density of a random variable $\mu \sim \mathcal{N}(x, 1)$ and it follows that $Z = (\mu - x) \sim \mathcal{N}(0, 1)$. So, we can write

$$\int_0^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mu' - x)^2} d\mu' = \mathbb{P}(0 < \mu < +\infty) \tag{4.12}$$

$$= \mathbb{P}(-x < Z < +\infty) \tag{4.13}$$

$$= \lim_{c \to +\infty} \Phi(c) - \Phi(-x) \tag{4.14}$$

$$= 1 - \Phi(-x), \tag{4.15}$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard Normal distribution. Hence, the posterior distribution is given by

$$p(\mu|x) = \frac{e^{-\frac{1}{2}(\mu - x)^2}}{(1 - \Phi(-x))\sqrt{2\pi}} \mathbf{1}_{(0,+\infty)}(\mu), \tag{4.16}$$

in which we recognize the truncated Normal distribution, $\mathcal{N}_{trunc}(x, 1; 0, +\infty)$ with mean x, variance 1, lower bound 0 and upper bound $+\infty$. In Figure 4.1 the posterior of μ given three different observations $x = 1\%$, $x = 5\%$ and $x = 10\%$ is presented. Notice that we now indeed have a distribution of parameter μ, which concentrates around the observation. Additionally, the effect of the improper prior can clearly be seen for $x = 1\%$ as the density cuts off for $\mu < 0$.

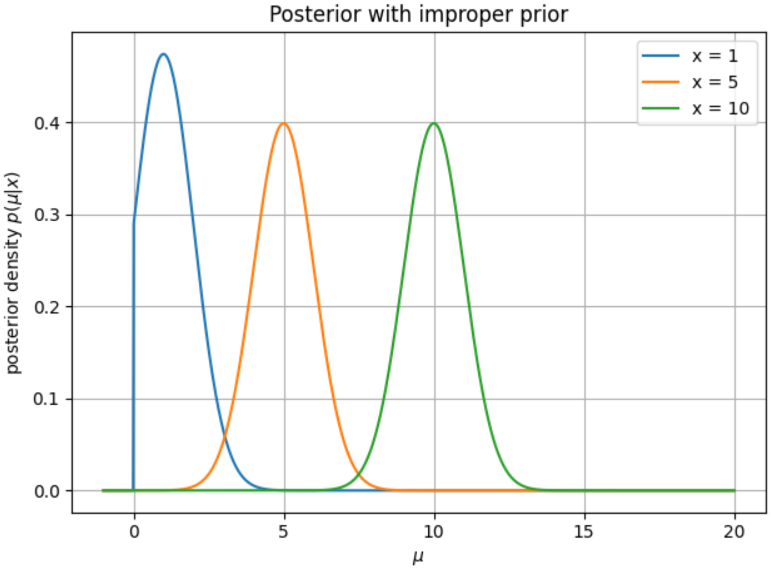


Figure 4.1: The posterior for our example with an improper prior $\mu \propto \mathbf{1}_{(0,+\infty)}$ and normally distributed observation x. In this example the posterior is presented in case that the observation is $x = 1$, $x = 5$ or $x = 10$ (Example 4.1).

Now, the DMO wants to use some estimator for μ. A frequentist DMO would, for example, use the maximum likelihood estimator, which is the value of μ that maximizes the likelihood. On the *contrary, a Bayesian DMO can compute the μ value that maximizes the posterior, also called the maximum a posteriori estimator (MAPE). For this example the MAPE is*

$$\hat{\mu}_{MAPE} = \underset{\mu \in (0,+\infty)}{\arg\max}\ p(\mu|x) \tag{4.17}$$

$$= x. \tag{4.18}$$

In Example 4.1 we have showed how to compute the posterior with an improper prior and one observation. In this example the posterior is indeed a valid probability distribution, even though we have chosen an improper prior. Moreover, as the improper prior gives little information, we see that the posterior concentrates around the observation. This is also the aim of the chosen improper prior, to let the observations have the most influence on the posterior. We only see the influence of the prior clearly for observations close to the bounds of the chosen domain of $\mu \in (0, +\infty)$. In addition, it is also interesting to see how the posterior behaves if we choose a prior that resembles a very certain knowledge about the parameter.

Example 4.2. *(Informative prior). Consider the same setting as in Example 4.1. However, instead of choosing an improper prior that gives little information, we choose a prior that represents our very certain belief about the mean of the yield μ that it has to be around 1%. We assume $\mu \sim \mathcal{N}(m_0, s_0^2)$ with $m_0 = 1, s_0^2 = 0.0009$. Moreover, suppose that there are n observations $\boldsymbol{x} = (x_1, \ldots, x_n)$ with $x_i \sim \mathcal{N}(\mu, 1)$ i.i.d. for $i = 1, \ldots, n$ instead of one observation. Then, the likelihood and prior are defined as*

$$p(\mu) = \frac{1}{s_0\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x - m_0}{s_0}\right)^2\right\}, \tag{4.19}$$

$$p(\boldsymbol{x}|\mu) = \frac{1}{(2\pi)^{n/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right\}. \tag{4.20}$$

Notice that this prior is also a conjugate prior, meaning that the posterior is also a Normal distribution. Define $n\bar{x} = \sum_{i=1}^{n} x_i$. We compute the posterior as follows, where we denote proportionality with respect to parameter μ as just $\propto$.

$$p(\mu|\boldsymbol{x}) \propto p(\boldsymbol{x}|\mu)p(\mu) \tag{4.21}$$

$$\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right\} \exp\left\{-\frac{1}{2}\left(\frac{\mu - m_0}{s_0}\right)^2\right\} \tag{4.22}$$

$$\propto \exp\left\{-\frac{1}{2s_0^2}\left(s_0^2\left(\sum_{i=1}^{n} x_i^2\right) - 2\mu s_0^2 n\bar{x} + s_0^2 n\mu^2 + \mu^2 - 2\mu m_0 + m_0^2\right)\right\} \tag{4.23}$$

$$\propto \exp\left\{-\frac{1}{2s_0^2}\left(\mu^2(1 + ns_0^2) - 2\mu\left(s_0^2 n\bar{x} + m_0\right)\right)\right\} \tag{4.24}$$

$$= \exp\left\{-\frac{1}{2s_0^2(1 + ns_0^2)^{-1}}\left(\mu - \left(\left(s_0^2 n\bar{x} + m_0\right)(1 + ns_0^2)^{-1}\right)\right)^2\right\}, \tag{4.25}$$

from which we conclude that the posterior is normally distributed as $\mu|\boldsymbol{x} \sim \mathcal{N}\left(m', (s')^2\right)$ *with*

$$m' = \frac{m_0 + s_0^2 \sum_{i=1}^n x_i}{1 + ns_0^2}, \tag{4.26}$$

$$(s')^2 = \frac{s_0^2}{1 + ns_0^2}. \tag{4.27}$$

Notice that for a very small s_0^2*, which means we are very certain of our belief,* $m' \approx m_0$ *and* $(s')^2 \approx s_0^2$*. This means the data should be very convincing of* μ *having some other value than the prior mean* m_0*, such that the term* $\sum_{i=1}^n x_i$ *"outweighs"* s_0^2*. In order to show the effect of the prior, we present the posterior for four different sets of observations. For simplicity, we consider the observations* $x_1 = 7\%$*,* $x_1 = \cdots = x_{100} = 7\%$*,* $x_1 = \cdots = x_{1000} = 7\%$ *and* $x_1 = \cdots = x_{10000} = 7\%$*. In Figure 4.2 the posterior for the several observations is shown. We can immediately see that it takes a lot of observations that counter the prior belief for the posterior to move to the actual observation. In addition, we see that the posterior "inherits" the certainty from the prior, since the posterior variance* $(s')^2$ *is not influenced by the observations.*

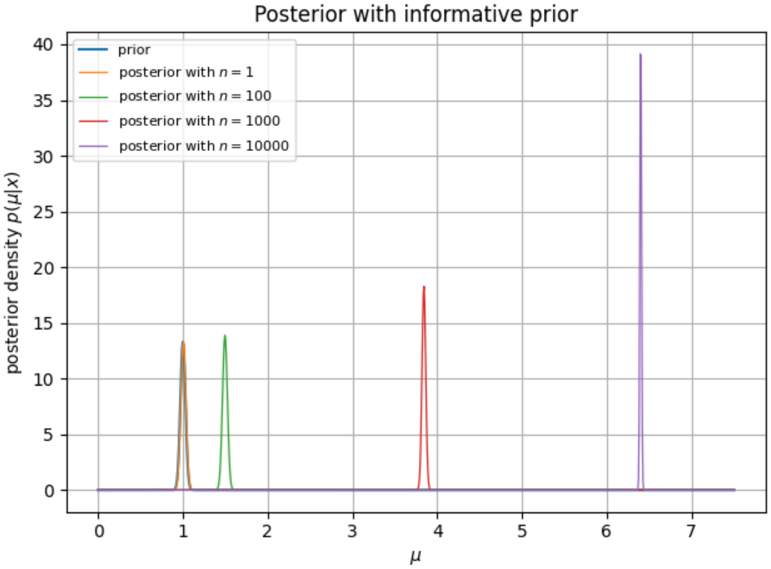


Figure 4.2: The posterior distribution for observations $x_1 = \cdots = x_n = 7\%$ with four scenario's $n = 1, 100, 1000, 10000$ (Example 4.2).

4.1.2 Posterior Predictive Distribution

In the previous subsection we discussed how to estimate parameters in the Bayesian approach. However, sometimes we are also interested in predictions of a new future observation $\tilde{x}$ given the already known observations $\boldsymbol{x} = (x_1, \ldots, x_n)$. In other words, we are interested in a predictive distribution. Gelman et al. (2014) explain the posterior predictive distribution, or posterior predictive in short, by first explaining its prior counterpart, the prior predictive distribution. This is the distribution *before* we have seen any previous data and is defined as

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\theta')p(\theta')d\theta'. \tag{4.28}$$

The prior predictive distribution turns out to be the marginal distribution of x. It is the *prior* predictive distribution, because the computation of the integral needs the prior and $\boldsymbol{x}$ is not conditioned on previous data. In contrast, the posterior predictive is the distribution of some not yet known observation $\tilde{x}$ *after* we have seen observations $\boldsymbol{x}$ with $\tilde{x}$ being an independent future observation. Analogous to the prior predictive distribution, the posterior predictive is defined as (Gelman et al., 2014, p. 7)

$$p(\tilde{x}|\boldsymbol{x}) = \int p(\tilde{x}, \theta|\boldsymbol{x})d\theta \tag{4.29}$$

$$= \int p(\tilde{x}|\theta)p(\theta|\boldsymbol{x})d\theta. \tag{4.30}$$

In this case, the posterior is needed for the computation of the *posterior* predictive distribution. Notice that $p(\tilde{x}|\theta)$ is the likelihood of observation $\tilde{x}$, which is distributed the same as $x_1, \ldots, x_n$. In the case that we have sequential data, like a time series $X_t := (x_1, \ldots, x_t)$, the posterior predictive translates naturally to forecasting. In particular, if we assume $x_1, \ldots, x_t$ to be i.i.d., the posterior predictive of a future observation x_{t+1} is defined analogously as $\tilde{x}$ as

$$p(x_{t+1}|X_t) = \int p(x_{t+1}, \theta|X_t) \tag{4.31}$$

$$= \int p(x_{t+1}|\theta)p(\theta|X_t)d\theta. \tag{4.32}$$

For complex models with more involved (in)dependence structures, the joint conditional distribution $p(x_{t+1}, \theta|X_t)$ may require a more involved decomposition as well (as we will see in Chapter 6). Moreover, the use of improper priors or priors that are not conjugate can result into probability distributions that are non-standard and cannot be expressed analytically. In such cases the posterior predictive needs to be simulated, which we will discuss later in this subsection. First, we elaborate on the posterior predictive in Example 4.3, where the posterior is the one in Example 4.2 and can be derived in closed-form due to the conjugate prior.

Example 4.3. *Consider the same setting as in Example 4.2. In that example we derived that the posterior is Normally distributed as* $\mu|\boldsymbol{x} \sim \mathcal{N}\left(m', (s')^2\right)$ *with*

$$m' = \frac{m_0 + s_0^2 \sum_{i=1}^n x_i}{1 + ns_0^2}, \tag{4.33}$$

$$(s')^2 = \frac{s_0^2}{1 + ns_0^2}, \tag{4.34}$$

where $\boldsymbol{x} = (x_1, \dots, x_n)$ are observations. Now, suppose that the observations are some time series $X_t := (x_1, \dots, x_t)$ of the monthly yields, where $x_i \sim \mathcal{N}(\mu, 1)$ i.i.d. Then, Gelman et al. (2014, p. 41) show that the posterior predictive is also a Normal distribution by using the mean and variance of the posterior and the likelihood in the case of one observation. However, in the same way we can derive the posterior predictive for multiple observations. By writing out $\mathbb{E}[x_{t+1}|X_t]$ and $\mathrm{Var}[x_{t+1}|X_t]$, the integral over the product of two Normal distributions reduces to computing

$$\begin{aligned}
\mathbb{E}[x_{t+1}|X_t] &= \mathbb{E}[\mathbb{E}[x_{t+1}|\mu, X_t]|X_t] && (4.35)\\
&= \mathbb{E}[\mathbb{E}[x_{t+1}|\mu]|X_t] && (4.36)\\
&= \mathbb{E}[\mu|X_t] && (4.37)\\
&= m', && (4.38)\\
\mathrm{Var}[x_{t+1}|X_t] &= \mathbb{E}[\mathrm{Var}[x_{t+1}|\mu, X_t]|X_t] + \mathrm{Var}[\mathbb{E}[x_{t+1}|\mu, X_t]|X_t] && (4.39)\\
&= \mathbb{E}[\mathrm{Var}[x_{t+1}|\mu]|X_t] + \mathrm{Var}[\mathbb{E}[x_{t+1}|\mu]|X_t] && (4.40)\\
&= \mathbb{E}[1|X_t] + \mathrm{Var}[\mu|X_t] && (4.41)\\
&= 1 + (s')^2. && (4.42)
\end{aligned}$$

Hence, $x_{t+1}|X_t \sim \mathcal{N}(m', 1 + (s')^2)$. Notice that the posterior predictive distribution of x_{t+1} is very similar to the posterior distribution $\mu|X_t$, as they have the same mean m'. In regard to the variance, though, the posterior predictive has an additional term 1, which is the variance of the distribution we assumed on the observations. This can be seen as the additional uncertainty that is involved with the prediction next to the uncertainty of the parameter. So, one can imagine that the posterior predictive looks like the posterior distribution, but with less concentration around the mean. In Figure 4.3 the posterior predictive is shown for x_{t+1} given four different scenario's of observation, which are the same as in Example 4.2. We can immediately see that the densities are much wider than in Figure 4.2, confirming the computation for the variance. Notice that since the prior is very concentrated around 1, this also works through the posterior predictive via the posterior.

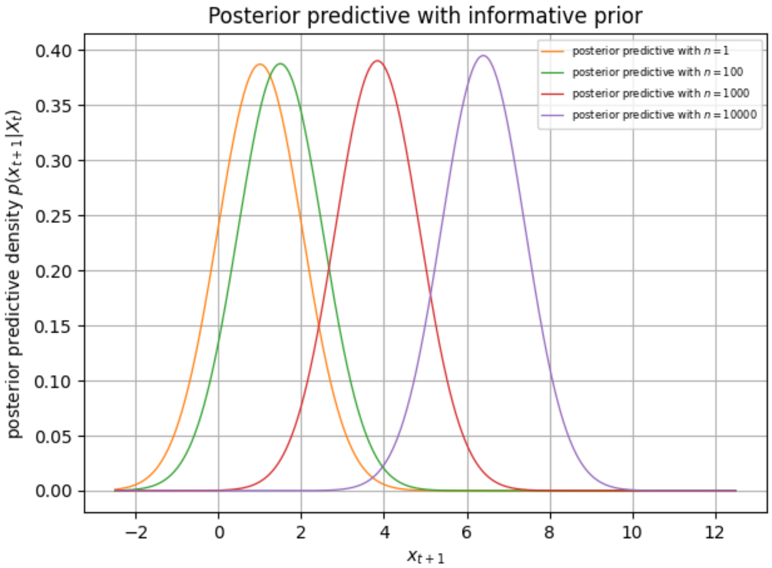


Figure 4.3: The posterior predictive distribution for observations $x_1 = \cdots = x_n = 7\%$ with four scenario's $n = 1, 100, 1000, 10000$ (Example 4.3).

It is not always possible to derive the integral in (4.29) in closed-form. In such cases we have to simulate the posterior predictive. If the posterior is known, this should not be a difficult task as the likelihood is known by assumption. Let $\boldsymbol{x} = (x_1, \dots, x_n)$ be n observations, where $x_i \sim q(\theta)$ i.i.d. and $q(\theta)$ is some known distribution. Suppose that the posterior $p(\theta|\boldsymbol{x})$ is known as well, but $p(\tilde{x}|\boldsymbol{x})$ cannot be derived analytically. Then, we can simulate the posterior predictive by random sampling, which is the simulation algorithm described in Algorithm 1. In Example 4.4 we show how the posterior predictive can be simulated.

Algorithm 1: Simulation for the posterior predictive distribution.

```
Input  : Observations $\boldsymbol{x} = (x_1, \dots, x_n)$ with $x_i \sim q(x_i; \theta)$ and known posterior $p(\theta|\boldsymbol{x})$.
Result: Posterior predictive samples $\tilde{x}^{(1)}, \dots, \tilde{x}^{(S)}$.
for $s = 1$ to $S$ do
    Sample $\theta^{(s)} \sim p(\theta|\boldsymbol{x})$;
    Sample $\tilde{x}^{(s)} \sim q(\tilde{x}; \theta^{(s)})$;
```

Example 4.4. *Consider the same setting as in Example 4.1. Recall that we have one observation $x \sim \mathcal{N}(\mu, 1)$ and that we choose the improper prior $p(\mu) \propto \mathbf{1}_{(0,+\infty)}(\mu)$. Consequently, the posterior is a truncated Normal distribution, so $\mu|x \sim \mathcal{N}(x, 1, 0, +\infty)$ as in (4.16). This means we have to compute the following integral to obtain the posterior predictive distribution for a new independent observation $\tilde{x}$.*

$$p(\tilde{x}) = \int p(\tilde{x}|\mu')p(\mu'|x)d\mu' \tag{4.43}$$

$$= \int_0^{+\infty} \frac{e^{-\frac{1}{2}(\tilde{x}-\mu')^2}}{\sqrt{2\pi}} \frac{e^{-\frac{1}{2}(\mu'-x)^2}}{(1-\Phi(-x))\sqrt{2\pi}} d\mu'. \tag{4.44}$$

Unlike Example 4.3, where we had two Normal distributions integrating to another Normal distribution, we cannot express the posterior predictive in (4.44) in closed-form as another probability distribution. However, to see how $\tilde{x}|x$ is distributed we can use Algorithm 1, which means that we have to sample for $s = 1, \ldots, S$

$$\mu^{(s)} \sim \mathcal{N}_{trunc}(x, 1, 0, +\infty), \tag{4.45}$$

$$\tilde{x}^{(s)} \sim \mathcal{N}(\mu^{(s)}, 1). \tag{4.46}$$

We will do this $S = 100000$ times for observation $x = 1$, $x = 5$ and $x = 10$. The results of the simulation are presented in Figure 4.4. We can see that most simulated values for $\tilde{x}$ are concentrated around x. Since $x \sim \mathcal{N}(\mu, 1)$ and $\mu > 0$, the simulated values of $\tilde{x}$ can become negative, but the average of those simulations cannot become negative.

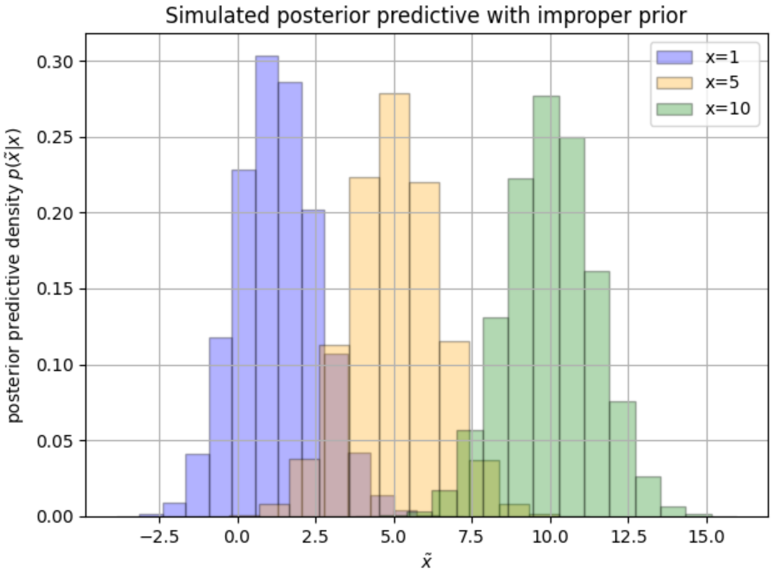


Figure 4.4: The simulation of the posterior predictive distribution in case of $x = 1$, $x = 5$ or $x = 10$ (Example 4.4).

4.2 Markov Chain Monte Carlo Methods

In the previous section we have seen that sometimes the posterior predictive distribution cannot be expressed analytically, so we have to resort to random sampling. As we will see in Chapter

6, in many practical situations even the posterior distribution cannot be expressed in closed-form. This means that we need a method to approximate the posterior with simulations. This is where *Markov Chain Monte Carlo* methods, or MCMC in short, come into play. In a nutshell, as the name suggests MCMC methods belong to a type of simulation techniques that are based on generating Markov chains by simulating a lot of samples (Monte Carlo). Specifically, Robert and Casella (2004, p. 268) define a MCMC method as "*any method for the simulation of a distribution f producing an* ***ergodic Markov chain*** $\left(X^{(t)}\right)$ *whose* ***stationary distribution*** *is f*". This definition can be broken down into three key concepts — Markov chains, ergodicity and the stationary distribution. We will elaborate on these three concepts.

A *Markov chain* is a sequence of random variables $\left(X^{(t)}\right) = X^{(0)}, X^{(1)}, \dots, X^{(t)}$ if for any $i = 0, \dots, t$ it holds that

$$\mathbb{P}(X_{i+1} = x | X_0 = x_0, \dots, X_i = x_i) = \mathbb{P}(X_{i+1} = x | X_i = x_i), \tag{4.47}$$

where $x_0, \dots, x_t$ are realizations of the random variables $X_0, \dots, X_t$. The expression in (4.47) is also called the *Markov property.* We have already seen an example of a Markov chain in Chapter 3. Recall that the $AR(1)$ process (see Figure 3.1) is characterized as a process where the state variable at the current time $\boldsymbol{x}_t$ only depends on the state at the previous time $\boldsymbol{x}_{t-1}$. In many practical cases a Markov chain is quite a restrictive assumption for a real-life process. However, it still allows for some indirect influence of the process through time as opposed to the assumption of i.i.d. random variables.

Subsequently, the second concept of *ergodicity* is quite a technical one. So, we will not provide the definition given by Robert and Casella (2004), but discuss the notion of ergodicity. In particular, an ergodic Markov chain is *irreducible* and *aperiodic.* Essentially, irreducibility means that the Markov chain $\left(X^{(t)}\right)$ can reach some point x' from any starting point $X_0 = x_0$ with positive probability (Robert and Casella, 2004, p. 231). So, it should not matter where the Markov chain begins as every other point can be reached if we "wait" long enough — but not infinitely long. Moreover, aperiodicity means that the Markov chain does not return to some point with a fixed pattern and does not "get stuck" at some point. Together, irreducibility and aperiodicity are sufficient to ensure that a Markov chain explores the entire space in a finite amount of time.

The third concept of reaching the *stationary distribution* f, is the goal of a MCMC method. Specifically, the stationary distribution f of a Markov chain $\left(X^{(t)}\right)$ is a distribution such that $X_t \sim f$ implies that $X_{t+1} \sim f$. This is the distribution that the Markov chain converges to and through the way that the Markov chain is constructed the stationary distribution is the same as the target distribution if the MCMC is run long enough.

4.2.1 Random Walk Metropolis

In this subsection we will discuss the MCMC method that is used for parameter estimation in our research, the Random Walk Metropolis algorithm or RWM in short. Suppose we have some posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ with a multidimensional parameter $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d) \in D \subset \mathbb{R}^d$ and observations $\boldsymbol{x} = (x_1, \dots, x_n)$. We denote the target distribution, which in this case is the posterior, as $P(\boldsymbol{\theta}) := p(\boldsymbol{\theta}|\boldsymbol{x})$. Then, the Random Walk Metropolis algorithm explores the parameter space D by constructing a random walk $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots$ by iteratively drawing a proposal $\boldsymbol{\theta}'$ from a symmetric *proposal density* $q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})$ and accepting or rejecting $\boldsymbol{\theta}'$ based on an acceptance rule.

Consequently, a Normal distribution is often chosen as symmetric proposal density. Then, the random walk with a Normal distribution is constructed as follows.

$$\begin{aligned} \boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + Z, \ Z \sim \mathcal{N}(\mathbf{0}, I_d\sigma^2), && (4.48) \\ &\overset{d}{=} \mathcal{N}(\boldsymbol{\theta}^{(t)}, I_d\sigma^2), && (4.49) \end{aligned}$$

where $\mathbf{0} \in \mathbb{R}^d$, $I_d \in \mathbb{R}^{d\times d}$ is the d-dimensional identity matrix and the scale is denoted by σ. In particular, the scale can be seen as some "step size" of how far the typical proposal can be from the last element of the random walk. Subsequently, the proposal $\boldsymbol{\theta}'$ gets either accepted and becomes the new element of the random walk $\boldsymbol{\theta}^{(t+1)}$, or gets rejected and a new proposal $\boldsymbol{\theta}' \sim \mathcal{N}(\boldsymbol{\theta}^{(t)}, I_d\sigma^2)$ is generated again. Important to note is that when the proposal is rejected, the last element of the random walk becomes the new last element again, so $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$. Now, the key of the RWM algorithm is that it accepts every proposal $\boldsymbol{\theta}'$ that is "more likely" than the last element of the random walk $\boldsymbol{\theta}^{(t)}$ based on the target distribution $P(\boldsymbol{\theta})$, but does not reject a proposal that is "less likely" beforehand. Instead, the proposal can still be accepted, but the less likely the proposal the less likely it will be that the proposal is accepted. This acceptance rule is formalized by defining the so-called *acceptance ratio* α as

$$\alpha = \min\left(1, \frac{P(\boldsymbol{\theta}')}{P(\boldsymbol{\theta}^t)}\right) \in [0,1], \tag{4.50}$$

and drawing some random uniformly distributed number $u \sim \mathcal{U}(0,1)$, where we accept the proposal $\boldsymbol{\theta}'$ if $u < \alpha$ and reject $\boldsymbol{\theta}'$ if $u > \alpha$. Notice that this acceptance rule has precisely the behaviour that we want. Consider the case that $P(\boldsymbol{\theta}') > P(\boldsymbol{\theta}^t)$. Then the fraction in (4.50) will be larger than one, which results into $\alpha = 1$. An acceptance ratio $\alpha = 1$ means that no matter the value of $u \in (0,1)$, $u < \alpha$ will always hold and $\boldsymbol{\theta}'$ is always accepted. Next, consider the case that $P(\boldsymbol{\theta}') < P(\boldsymbol{\theta}^t)$, then the smaller $P(\boldsymbol{\theta}')$ is, the smaller α will be. A very small α means that the probability of $u < \alpha$ is also very small, which results into a small probability of accepting $\boldsymbol{\theta}'$.

The choice of the proposal density together with the definition of the acceptance ratio ensures that the random walk explores the parameter space D, while it spends most time in the regions of D that have a high probability according to $P(\boldsymbol{\theta})$. Together, they ensure that the generated Markov chain $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(t)}$ — the random walk in this case — is ergodic and for large enough t converges to the desired target distribution P. The RWM algorithm can be summarized as follows in Algorithm 2. Notice that the provided RWM algorithm has a multidimensional scale $\boldsymbol{\sigma} \in \mathbb{R}^d$, which is adjustable for each parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ resulting into the variance for q given by $I_d\boldsymbol{\sigma}^2 = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_d^2)$.

Algorithm 2: Random Walk Metropolis

```
Input  : Starting point $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^d$, scale $\boldsymbol{\sigma} \in \mathbb{R}^d$ and number of iterations $T$
Result: Sequence of accepted states: $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, ..., \boldsymbol{\theta}^{(T)} \overset{d}{\sim} P(\boldsymbol{\theta})$
for $t = 1$ to $T$ do
    Generate a proposal state $\boldsymbol{\theta}' \sim \mathcal{N}(\boldsymbol{\theta}^{(t)}, I_d\boldsymbol{\sigma}^2) \overset{d}{=} q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})$;
    Compute the acceptance ratio: $\alpha = \min\left(1, \frac{P(\boldsymbol{\theta}')}{P(\boldsymbol{\theta}^{(t)})}\right)$;
    Generate a uniform random number: $u \sim \mathcal{U}(0,1)$;
    if $u < \alpha$ then
        Accept the proposed state;
        $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}'$;
    else
        Reject the proposed state;
        $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$;
```

Remark. *It can happen that the log-posterior is needed instead of the posterior if, for example, the log-likelihood is already known and obtaining the likelihood is numerically not possible. In which case, the RWM algorithm can be used in a slightly adjusted way. Instead of α we compute the log-transformation* $\log(\alpha)$*. In particular, we have*

$$\log(\alpha) = \min\left(\log(1), \log\left(\frac{P(\boldsymbol{\theta}')}{P(\boldsymbol{\theta}^{(t)})}\right)\right) \tag{4.51}$$

$$= \min\left(0, \log(P(\boldsymbol{\theta}')) - \log\left(P\left(\boldsymbol{\theta}^{(t)}\right)\right)\right), \tag{4.52}$$

for which $u < \alpha$ implies that $\log(u) < \log(\alpha)$*.*

Furthermore, we note that the Random Walk Metropolis algorithm is actually a special case of the Metropolis-Hastings algorithm. Specifically, the more general Metropolis-Hastings algorithm does not require the proposal density q to be symmetric. As a consequence, the acceptance ratio for the RWM algorithm in (4.50) is a simplification of the so-called *detailed balance* condition the Markov chain has to satisfy. Essentially, the detailed balance condition means that the probability of the Markov chain moving from some point x to another point y has to be the same as moving from y to x (Robert and Casella, 2004, p. 230). This condition results into the more general acceptance ratio of the Metropolis-Hastings algorithm

$$\alpha = \min\left(1, \frac{P(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}')}{P(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})}\right). \tag{4.53}$$

Then, a symmetric proposal density q results into the acceptance ratio as in (4.50) since $q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}') = q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})$.

4.2.2 Convergence of the Markov Chains

When running the RWM algorithm there are essentially two input variables we can vary, the scale $\boldsymbol{\sigma}$ and the starting point $\boldsymbol{\theta}^{(0)}$. In this section we elaborate on how the two input variables affect the convergence of the Markov chains $\left(\boldsymbol{\theta}^{(T)}\right)$ and how we can evaluate whether the chains have converged to the stationary distribution P.

Influencing Convergence with Starting Point and Scale

First, recall that because of ergodicity of the Markov chains, that is the independence of starting points, the starting point does not matter *in theory.* However, it does affect the speed of convergence. If the starting point is in a low density region of the target distribution, then the chains have to explore the parameter space longer until they find the higher density regions. These first iterations for exploring are called the *burn-in* period. In contrast, if the starting point is already at a high density region of the target distribution, the burn-in period will not be as long or not necessary. So, even though ergodicity guarantees convergence from any starting point, *in practice* a good starting point $\boldsymbol{\theta}^{(0)}$ will result into faster convergence. According to Young and Smith (2005) the maximum likelihood estimator is often used as starting point. Given the little to no initial knowledge about the posterior the MLE can serve as a practical choice.

Then, the scale $\boldsymbol{\sigma}$ is perhaps the most important input variable to affect of convergence. Specifically, a scale can affect the convergence either by being too small or too large. A scale that is too large will result into proposals $\boldsymbol{\theta}'$ that lay in a low density region of P and will be rejected often (small α). One can imagine that as soon as a proposal in a high density region is accepted, the proposals from the new element of the Markov chain will often not be near the accepted proposal, but again in a low density region. On the contrary, a scale that is too small will accept proposals often (large α), because a proposal will often be *relatively* better than the last element of the Markov chain. However, the chain will improve too slowly with every iteration resulting into slow convergence. A good scale should balance exploring the low density regions and finding the high density regions. Roberts and Rosenthal (2001) show that the optimal scale should result into an average acceptance ratio of $\bar{\alpha} \approx 0.234$, where $\bar{\alpha}$ is defined as

$$\bar{\alpha} := n^{-1}\{\text{accepted proposals}\}. \tag{4.54}$$

In practice, an average acceptance ratio $\bar{\alpha}$ between 0.1 and 0.4 will provide good convergence as well. We show the significance of the choice for both the starting point and the scale in Example 4.5.

Example 4.5. *Consider a random variable $\boldsymbol{X} = (X_1, X_2)$ that is distributed according to a bivariate Normal distribution, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, I_2)$, with known mean $\boldsymbol{\mu} = [5,5]^T$ and variance I_2. Then, the target distribution $P(\boldsymbol{x})$ is defined as*

$$P(\boldsymbol{x}) = (2\pi)^{-1} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{x}-\boldsymbol{\mu})\right\}. \tag{4.55}$$

We run the RWM algorithm with $T = 10000$ iterations to approximate P for x_1 and x_2 for three different cases. The first one is with a starting point and a scale such that the convergence is fast and the average acceptance ratio is between 0.1 and 0.4. The second case is with a starting point in a low density region and a small scale. Finally, the third case is again with a starting point in a low density region, but with a large scale.

1. *A starting point at the mean $\boldsymbol{x}^{(0)} = [5,5]^T$ in a high density region and scale $\boldsymbol{\sigma} = [1.5, 1.5]^T$;*

2. A starting point at $\boldsymbol{x}^{(0)} = [20, -10]^T$ *in a low density region and a small scale* $\boldsymbol{\sigma} = [0.01, 0.01]^T$*;*

3. A starting point at $\boldsymbol{x}^{(0)} = [20, -10]^T$ *in a low density region and a large scale* $\boldsymbol{\sigma} = [7, 7]^T$*;*

The results are presented in Figure 4.6 as trace plots of the Markov chains of x_1 *and* x_2 *and the average acceptance ratio* $\bar{\alpha}$ *is provided in Table 4.5. The average acceptance ratio's are as expected for each case and are unsurprising if we look at the trace plot of the chains for* x_1 *and* x_2*. If a chain converges it can be seen in the trace plot as it reaches some value and progresses as white noise around that value. In Figure 4.6a we can see that this is directly the case and the trace plots indeed resemble a white noise. In the case of a starting point that is in a low density region with a small scale we see in Figure 4.6b that after 10000 iterations the chains are still not even close to the mean* μ*, around which most of the density is concentrated. Then, for the last case we can see in Figure 4.6c that the chains reach the high density region quite fast, but the proposal density struggles to consistently generate a new proposal that is close to the last element of the chain.*

Table 4.5: The average acceptance ratio for the three different scenario's of the RWM run of Example 4.5.

Scenario	$\bar{\alpha}$
High density $\boldsymbol{x}^{(0)}$, "optimal" $\boldsymbol{\sigma}$	0.3963
Low density $\boldsymbol{x}^{(0)}$, small $\boldsymbol{\sigma}$	0.9382
Low density $\boldsymbol{x}^{(0)}$, large $\boldsymbol{\sigma}$	0.0343

Convergence Diagnostics

In Example 4.5 we have seen that the convergence can be influenced with the starting point and the scale, where we looked at the average of the acceptance ratio's and trace plots. Besides visually evaluating convergence, we can statistically test the convergence as well. Intuitively, one would expect that when a Markov chain reaches the stationary distribution it would not show some trend to another value anymore. Geweke (1991) presents a convergence diagnostic, also called the *Geweke diagnostic*, that essentially tests whether the first segment and the last segment of the chain have significantly different means. In particular, consider a Markov chain $\left(X^{(T)}\right)$. Then, the Geweke diagnostic is a test to assess whether the mean of the first segment of the Markov chain $X^{(0)}, \ldots, X^{(T_A)}$ and the last segment $X^{(T_B)}, \ldots, X^{(T)}$ are significantly different or the same. The Geweke diagnostic and the test based on this statistic are given in Definition 4.6 and Theorem 4.7, which are based on Geweke (1991, pp. 6-7) and Robert and Casella (2004, pp. 508-509) respectively.

Definition 4.6. *(Geweke diagnostic). Let* $\left(X^{(T)}\right) = X^{(0)}, \ldots, X^{(T)}$ *be a sequence of observations. Let* $T_A = T\tau_A$ *and* $T_B = T\tau_B$*, where* τ_A, τ_B *with* $\tau_A + \tau_B < 1$ *are the first and last portions of the sequence. Then, the Geweke diagnostic is defined as*

$$G = \frac{\bar{X}_{T_A} - \bar{X}_{T_B}}{\sqrt{\frac{\sigma_A^2}{T_A} + \frac{\sigma_B^2}{T_B}}}, \tag{4.56}$$

where $\bar{X}_{T_A}$ *and* $\bar{X}_{T_B}$ *are defined as*

$$\bar{X}_{T_A} = T_A^{-1} \sum_{t=0}^{T_A} X^{(t)}, \tag{4.57}$$

$$\bar{X}_{T_B} = T_B^{-1} \sum_{t=T-T_B+1}^{T} X^{(t)}, \tag{4.58}$$

and σ_A^2 and σ_B^2 are defined as

$$\sigma_A^2 = T_A^{-1} \sum_{t=0}^{T_A} \left(\bar{X}_{T_A} - X^{(t)}\right)^2, \tag{4.59}$$

$$\sigma_B^2 = T_B^{-1} \sum_{t=T-T_B+1}^{T} \left(\bar{X}_{T_B} - X^{(t)}\right)^2. \tag{4.60}$$

Theorem 4.7. *Consider the Geweke diagnostic G as defined in Definition 4.6. Consequently, as $T \longrightarrow \infty$ it follows that $G \longrightarrow \mathcal{N}(0,1)$. So, for large T and some significance level α, we can test for*

$$\begin{cases} H_0: & \text{The means of the first and last segments of } \left(X^{(T)}\right) \text{ have no significant difference,} \\ H_A: & \text{The means of the first and last segments of } \left(X^{(T)}\right) \text{ have significant difference,} \end{cases} \tag{4.61}$$

by the following acceptance rule

$$\begin{cases} \text{Accept } H_0, & \text{if } |G| < z_{\alpha/2}, \\ \text{Reject } H_0, & \text{if } |G| > z_{\alpha/2}, \end{cases} \tag{4.62}$$

where $z_x = \Phi(x)$ with $\Phi(\cdot)$ the standard Normal cdf.

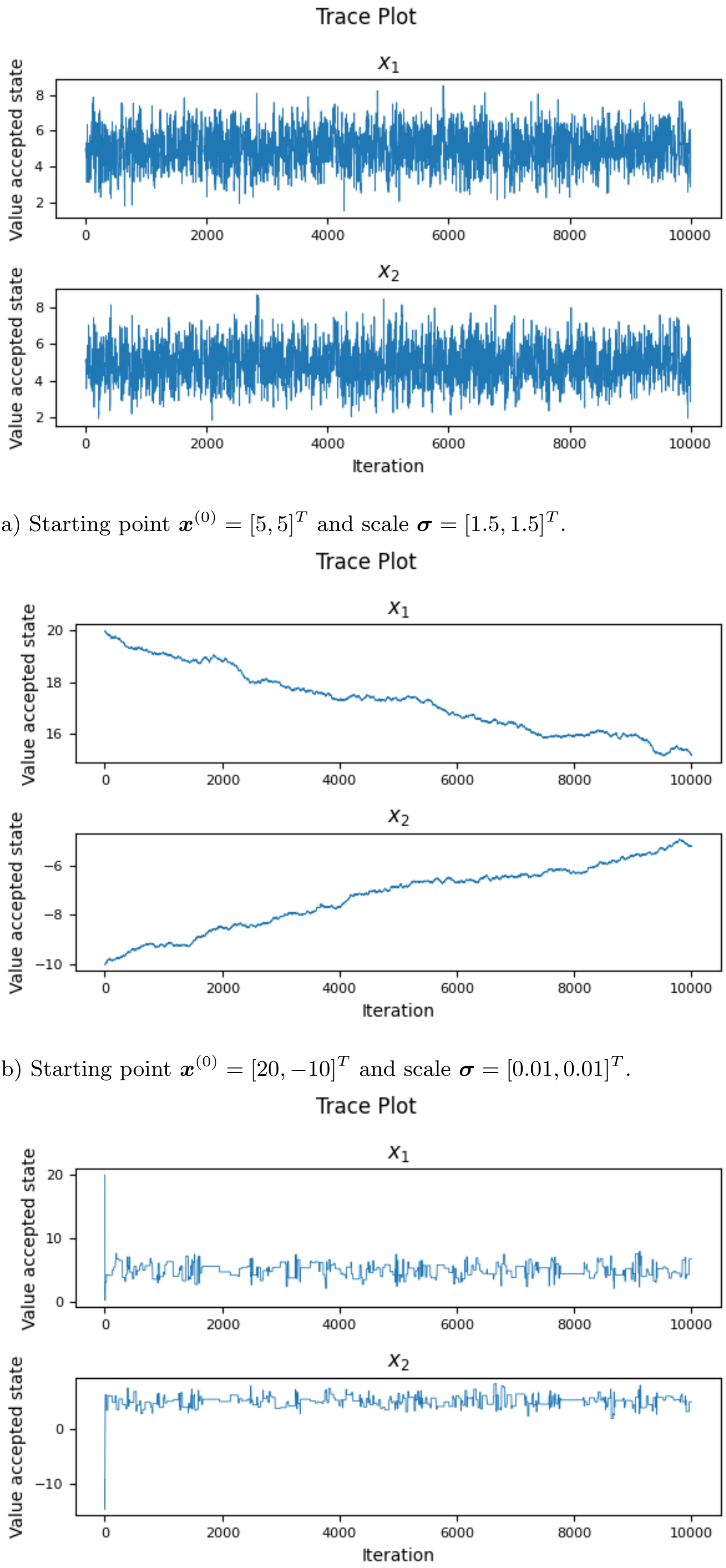
We note that for large T the terms σ_A^2 and σ_B^2 approximate the asymptotic variance of the Markov chain based on their respective subsample. The asymptotic variance is related to the spectral density, which is originally used by Geweke (1991). We refer to Robert and Casella (2004, p. 508) for the exact relationship between the asymptotic variance and the spectral density and we refer to Brockwell and Davis (1991, Section 4.4) for more detailed information on spectral analysis of time series in general as it goes beyond the scope of this chapter. Furthermore, Geweke (1991) suggests $\tau_A = 0.1$ and $\tau_B = 0.5$. Finally, in Example 4.8 we show how the Geweke diagnostic can be used in practice.

Example 4.8. *Consider the same setting as in Example 4.5. Recall that we looked at three different cases of a RWM run with $T = 10000$ iterations. Then, using $\tau_A = 0.1$ and $\tau_B = 0.5$ we obtain $T_A = 1000$ and $T_B = 5000$. Now, the Geweke diagnostic can be computed by calculating $\bar{x}_{i,T_A}, \bar{x}_{i,T_B}, \sigma_{i,A}^2$ and $\sigma_{i,B}^2$ for $i = 1, 2$. We use a significance level of $\alpha = 0.05$, so $z_{\alpha/2} \approx 1.96$. The results for the*

Markov chains of parameters x_1 and x_2 and the three scenario's are presented in Table 4.7. For the first two scenario's the Geweke test gives unsurprising results, since the trace plot already indicated a white noise for the chains in the first scenario and an obvious trend in the chains for the second scenario (Figure 4.6a and Figure 4.6b resp.). The results of the Geweke test for the last scenario might be surprising at first sight, but recall that this test only assesses whether the means of the first and last segment are significantly different. For the third scenario with bad convergence this is indeed the case (Figure 4.6c). This result shows that evaluating convergence is tricky and needs both visual methods and tests to assess it correctly.

Table 4.7: The results of the Geweke diagnostic G and test decision of Example 4.8.

Scenario	x_i	$\|G\|$	Accept/reject H_0
High density $\boldsymbol{x}^{(0)}$, "optimal" $\boldsymbol{\sigma}$	$i=1$	0.04862	Accept H_0
	$i=2$	0.04303	Accept H_0
Low density $\boldsymbol{x}^{(0)}$, small $\boldsymbol{\sigma}$	$i=1$	4.87621	Reject H_0
	$i=2$	5.87521	Reject H_0
Low density $\boldsymbol{x}^{(0)}$, large $\boldsymbol{\sigma}$	$i=1$	0.05936	Accept H_0
	$i=2$	0.22985	Accept H_0



(a) Starting point $\boldsymbol{x}^{(0)} = [5,5]^T$ and scale $\boldsymbol{\sigma} = [1.5, 1.5]^T$.

(b) Starting point $\boldsymbol{x}^{(0)} = [20,-10]^T$ and scale $\boldsymbol{\sigma} = [0.01, 0.01]^T$.

(c) Starting point $\boldsymbol{x}^{(0)} = [20,-10]^T$ and scale $\boldsymbol{\sigma} = [7, 7]^T$.

Figure 4.6: Trace plots of (x_1, x_2) for the three different cases of Example 4.5.

5

Literature on Modeling Yield Curves and Volatility

In this chapter we discuss our findings in the literature about yield curve modeling and volatility modeling. There are several methods to estimate the yield curve, but we will only focus on the so-called *Nelson-Siegel* model, of which a variant is used by the DSTA. Other methods to estimate the yield curve include *cubic splines* and *B-splines*, for which we refer to Filipovic (2009, Chapter 3). In Section 5.1 we discuss the original Nelson-Siegel model (C. R. Nelson and Siegel, 1987) and work towards the Dynamic Nelson-Siegel (DNS) model (Diebold et al., 2006), which is the starting point of our research in volatility modeling. In the subsequent section we discuss how to model volatility in the DNS framework and we provide some concluding remarks, on which we base our approach to modeling volatility.

5.1 Nelson-Siegel Yield Curves

The Nelson-Siegel model is a parametric estimation method, which means that the yield curve can be described with a finite set of parameters. In particular, a yield curve is estimated by the Nelson-Siegel model as

$$y_t(\tau) = \beta_{1,t} + \beta_{2,t}\left(\frac{1-e^{-\lambda_t\tau}}{\lambda_t\tau}\right) + \beta_{3,t}\left(\frac{1-e^{-\lambda_t\tau}}{\lambda_t\tau} - e^{-\lambda_t\tau}\right), \tag{5.1}$$

and can be described by the parameters $\{\beta_{1,t}, \beta_{2,t}, \beta_{3,t}, \lambda_t\}$, where $\beta_{i,t},\ i = 1, 2, 3$ are *latent variables* and λ is the *exponential decay parameter* (Diebold and Li, 2006, p. 341). The factors that are multiplied with $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ are called *loadings*. We notice that the definition of the Nelson-Siegel yield curve as in (5.1) is not the original definition of C. R. Nelson and Siegel (1987), but is a slightly modified but equivalent definition given by Diebold and Li (2006, p. 341). The formulation of Diebold and Li (2006) minimizes high correlation between $\beta_{2,t}$ and $\beta_{3,t}$ and is easier to interpret economically. We will first elaborate on the effect of the parameters $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ on the shape of the yield curve and afterwards we will discuss the effect of λ_t on the yield curve shape.

5.1.1 Interpretation of the Latent Variables

The latent variables $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ can be interpreted as the level, slope and curvature of the yield curve, which are associated with the long-term, short-term and medium-term maturities of the yield curve respectively. The interpretation for the parameters can be derived from (5.1). The long-term parameter $\beta_{1,t}$ is associated with the level of the yield curve, because a change in $\beta_{1,t}$ changes the yields with the same amount as the loading is equal to one. Additionally, by taking limits we obtain

$$\begin{aligned} \lim_{\tau \to +\infty} y_t(\tau) &= \lim_{\tau \to +\infty} \left\{ \beta_{1,t} + \beta_{2,t} \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} \right) + \beta_{3,t} \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau} \right) \right\} && (5.2) \\ &= \beta_{1,t}. && (5.3) \end{aligned}$$

Subsequently, the short term parameter $\beta_{2,t}$ is associated with the (downward) slope of the yield curve. Particularly, if the slope is defined as the linear difference between the yield at $\tau \downarrow 0$ and $\tau \to +\infty$, then

$$\begin{aligned} \lim_{\tau \downarrow 0} y_t(\tau) &= \lim_{\tau \downarrow 0} \left\{ \beta_{1,t} + \beta_{2,t} \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} \right) + \beta_{3,t} \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau} \right) \right\} && (5.4) \\ &= \beta_{1,t} + \beta_{2,t}, && (5.5) \end{aligned}$$

from which it follows that

$$\begin{aligned} \lim_{\tau \to +\infty} y_t(\tau) - \lim_{\tau \downarrow 0} y_t(\tau) &= \beta_{1,t} - (\beta_{1,t} + \beta_{2,t}) && (5.6) \\ &= -\beta_{2,t}. && (5.7) \end{aligned}$$

Finally, the interpretation of the medium-term parameter $\beta_{3,t}$ as curvature is a result of the corresponding loading converging to zero for both limits $\tau \downarrow 0$ and $\tau \to +\infty$ and becomes more obvious if we plot the loadings for each $\beta_{i,t}$, $i = 1, 2, 3$. The plot of the loadings is provided in Figure 5.1, where we see that the loading of $\beta_{3,t}$ indeed converges to zero for both ends. However, it has some "hump" for the medium-term maturities.

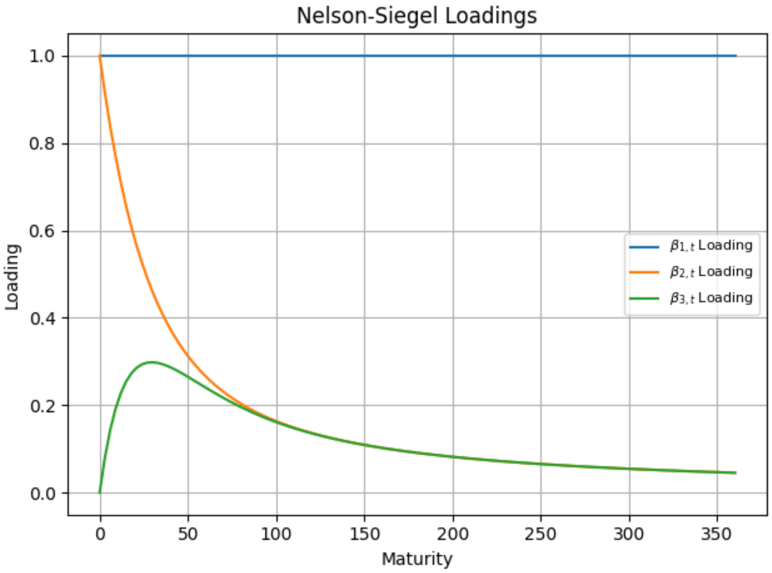


Figure 5.1: The loadings for $\beta_{i,t},\ i = 1, 2, 3$ for maturities $\tau \in (0, 360]$.

So, one can imagine that a linear combination of the loadings, where $\beta_{1,t}$ and $\beta_{2,t}, \beta_{3,t}$ are variable, result into different yield curve shapes. In Figure 5.2 we show some yield curves for different values of $\beta_{i,t},\ i = 1, 2, 3$, where we fixed two $\beta_{i,t}$'s and vary the third one. The values at which the two $\beta_{i,t}$'s are fixed are provided in the legends of the plots in Figure 5.2. For this specific example λ_t is fixed as $\lambda_t := \lambda = 0.0609$, which Diebold and Li (2006) use in their paper as well. We can see that the effect of $\beta_{1,t}, \beta_{2,t}$ on the level and slope respectively are as expected from (5.3) and (5.7), whereas $\beta_{3,t}$ seems to mostly affect the short-term and medium-term yields.

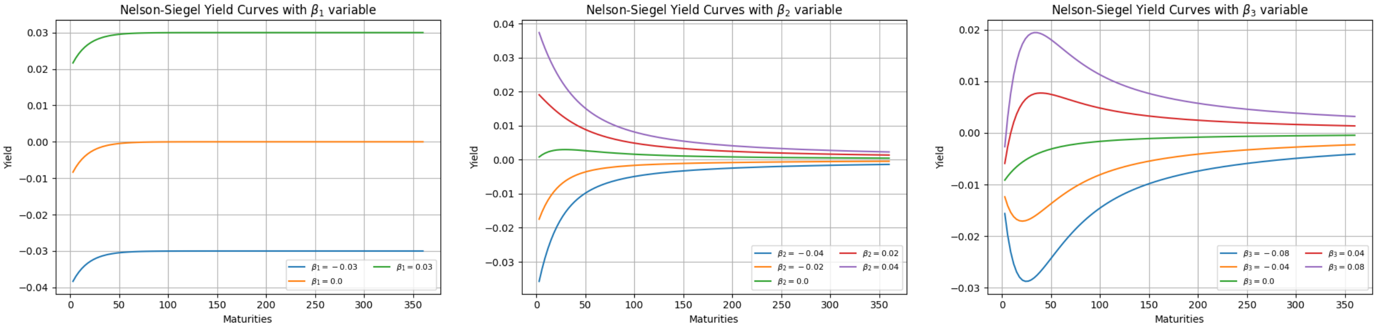


Figure 5.2: Different Nelson-Siegel yield curve shapes for a variable $\beta_{i,t},\ i = 1, 2, 3$ and fixed $\lambda = 0.0609$.

5.1.2 Interpretation of the Decay Parameter

The exponential decay parameter λ_t determines how well the Nelson-Siegel yield curve fits either the long maturity yields (small λ_t) or short maturity yield (large λ_t) (Diebold and Li, 2006, p. 341). This is due to the effect of λ_t on how fast the yield curve converges to the level. In Figure 5.3 a plot with variable λ_t and fixed $\beta_{i,t},\ i = 1, 2, 3$ is shown. We can see that λ_t determines whether the maximum of the $\beta_{3,t}$ loading is reached at shorter or longer maturities. Koopman et al. (2010) present a model with time-varying λ_t, which means that the loadings of the latent variables also change with time. However, we will not consider time-varying loadings in our research, so we refer to Koopman et al. (2010) for further discussion on such models.

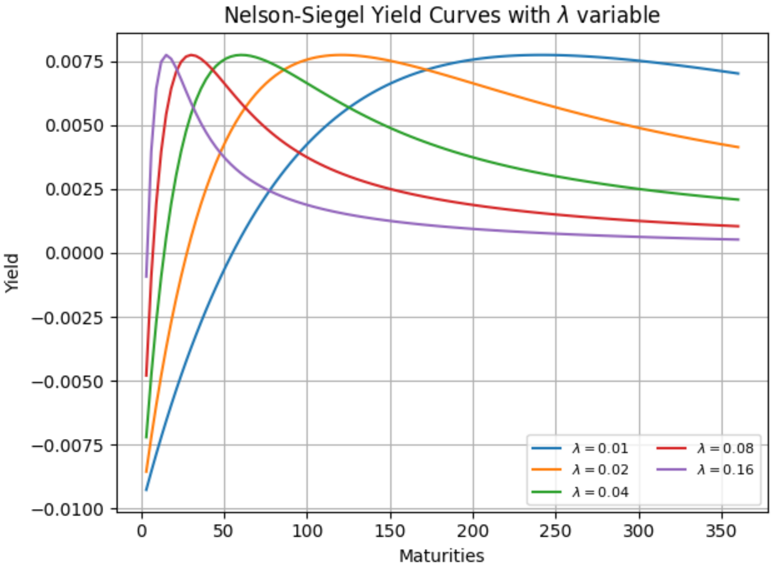


Figure 5.3: Different Nelson-Siegel yield curve shapes for a variable λ_t and fixed $\beta_{1,t} = 0, \beta_{2,t} = -0.01, \beta_{3,t} = 0.04$.

5.1.3 Dynamic Nelson-Siegel Model

The Dynamic Nelson-Siegel model is proposed in a paper by Diebold et al. (2006), in which they translate the original Nelson-Siegel model in a state-space framework. So, instead of treating the latent variables $\beta_{i,t},\ i = 1, 2, 3$ as some fixed parameter for each yield $y_t(\tau)$, the authors assume that the latent variables are state variables of a state-space model. This means that the latent variables, which are now state variables, follow a first order vector autoregressive process, denoted by VAR(1). So, the latent variables are a process on itself as well (and have *dynamics*) defined by the authors as

$$\begin{bmatrix}\beta_{1,t}-\mu_1\\ \beta_{2,t}-\mu_2\\ \beta_{3,t}-\mu_3\end{bmatrix} = \begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\\ \phi_{31} & \phi_{32} & \phi_{33}\end{bmatrix}\begin{bmatrix}\beta_{1,t-1}-\mu_1\\ \beta_{2,t-1}-\mu_2\\ \beta_{3,t-1}-\mu_3\end{bmatrix} + \begin{bmatrix}\eta_{1,t}\\ \eta_{2,t}\\ \eta_{3,t}\end{bmatrix}, \tag{5.8}$$

$$\Leftrightarrow \boldsymbol{\beta}_t - \boldsymbol{\mu} = \Phi(\boldsymbol{\beta}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\eta}_t, \tag{5.9}$$

where $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta)$. Subsequently, the authors relate the state variable $\boldsymbol{\beta}_t$ with the yields via the same loadings as the original Nelson-Siegel model. Suppose that for each time point t there are M yield observations for maturities $\tau_1, \ldots, \tau_M$, then the observation equation for the yields is given by

$$\begin{bmatrix}y_t(\tau_1)\\ \vdots\\ y_t(\tau_M)\end{bmatrix} = \begin{bmatrix}1 & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} - e^{-\lambda\tau_1}\\ \vdots & \vdots & \vdots\\ 1 & \frac{1-e^{-\lambda\tau_M}}{\lambda\tau_M} & \frac{1-e^{-\lambda\tau_M}}{\lambda\tau_M} - e^{-\lambda\tau_M}\end{bmatrix}\begin{bmatrix}\beta_{1,t}\\ \beta_{2,t}\\ \beta_{3,t}\end{bmatrix} + \begin{bmatrix}\varepsilon_{1,t}\\ \vdots\\ \varepsilon_{M,t}\end{bmatrix}, \tag{5.10}$$

$$\Leftrightarrow \boldsymbol{y}_t = \Lambda\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, \tag{5.11}$$

where $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon)$. Moreover, the authors notice that Σ_ε is often taken as a diagonal matrix, which means that the yield noise terms for different maturities are assumed to be uncorrelated. Intuitively, this means that it is assumed that the different segments of the yield curve have their own "dynamics" in the market. Additionally, this also simplifies the model significantly as it reduces the number of covariance parameters from $\frac{1}{2}M(M+1)$ to only M. On the contrary, according to the authors Σ_η is often assumed to be non-diagonal, which allows the underlying drivers of the yield curve, $\boldsymbol{\beta}_t$, to be correlated. Furthermore, it is also assumed that $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\eta}_t$ are independent of each other at time t. Then, the general DNS model is given by

$$\begin{cases}\boldsymbol{y}_t = \Lambda\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon),\\ \boldsymbol{\beta}_t - \boldsymbol{\mu} = \Phi(\boldsymbol{\beta}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta),\end{cases} \tag{5.12}$$

which satisfies Definition 3.1 of a state-space model. Diebold et al. (2006) recommend using the state-space approach instead of the method described in Diebold and Li (2006) since the state-space framework allows us to take into account the uncertainty of both the yield observations and the state variables. Moreover, we can use the Kalman filter described in Section 3.3 to estimate the state variables $\boldsymbol{\beta}_t$. Additionally, the state-space representation assumes that the state variables $\boldsymbol{\beta}_t$ and the yields $\boldsymbol{y}_t$ are stochastic processes. This enables extensions with, for example, stochastic volatility in the noise terms. However, using a state-space approach can also have some drawbacks. As we have seen in Section 3.3 the Kalman filter requires initial values for the state variable, $\hat{\boldsymbol{\beta}}_0^0$ and covariance matrix P_0^0. It could be possible that the estimates of $\boldsymbol{\beta}_t$ show strong dependency on the initial state values if the observation noise variance is large and, as a result, the observations do not contribute significantly to the Kalman gain ($K_t \approx 0$). Furthermore, the state-space variant of the Nelson-Siegel model has — potentially much — more parameters than the original model. So, there is a risk of overfitting, which means that it is possible that the model will not capture the actual dynamics of the bond yields outside the data well.

5.2 Modeling Yield Curve Volatility

In this section we discuss the different ways we can model volatility. In the financial context, volatility is a measure that describes the degree of variation of some asset price over time. It can be seen as some measure of uncertainty of the market. Oosterlee and Grzelak (2019, p. 28) describe volatility as a "*statistical measure of the tendency of an asset to rise or fall sharply within a period of time*". So, volatility gives some sense to what extend a process exhibits rapid or slow movements. A market with high volatility is expected to have larger price swings, which means that the prices deviate more from their mean value in a short period of time. On the contrary, low volatility suggests a more stable and predictable market environment, where prices do not deviate a lot from their mean over a longer period of time. So, we want to use a type of model that can capture such sharp movements. In the first subsection we discuss two popular types of such models. In the following subsection we discuss these standard volatility models in the context of the DNS framework.

5.2.1 Standard Volatility Models

When modeling volatility we generally have the choice between two popular types of volatility models. The first volatility model is the *Generalized Autoregressive Conditional Heteroskedasticity* model, or GARCH in short, originally introduced by Bollerslev (1986). The second volatility model is the *Stochastic Volatility* model, or SV in short, introduced by Taylor (1982). Both models are typically used to model the so-called *returns* or *relative gains* of an asset (Shumway and Stoffer, 2011, pp. 280-281), which is the percentage gain or loss of an asset price. The notion of *asset returns* is related to the idea of a *bond yield*, but it is mostly associated with stocks. Nevertheless, as we will see in the next subsection both models are also used to model volatility in bond yields.

GARCH Model

First, we discuss the standard GARCH model. Suppose the returns y_t of an asset are given by

$$y_t = \sigma_t \varepsilon_t, \tag{5.13}$$

where $\varepsilon_t \sim \mathcal{N}(0,1)$ and σ_t is the volatility, which is the standard deviation of y_t. Notice that this implies $y_t \sim \mathcal{N}(0, \sigma_t^2)$. Then, a GARCH$(p,q)$ model is defined as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{5.14}$$

with $\alpha_i > 0$ for $i = 0, \ldots, p$, $\beta_j > 0$ for $j = 1, \ldots, q$ and $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$. The *autoregressive* part refers to the fact that the return at time t, y_t, is dependent on the past return y_{t-1} via σ_t. Moreover, the *conditional heteroskedasticity* refers to the variance of the returns σ_t^2 changing over time, which are conditional on the past return y_{t-1}. Often a GARCH$(1,1)$ model is used, given by

$$\begin{cases} y_t = \sigma_t \varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0,1), \\ \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \end{cases} \tag{5.15}$$

Notice that the specification for the volatility σ_t as a GARCH process does not have some stochastic noise term. The only stochastic process in the definition of the GARCH volatility is the past return y_{t-1}^2, but the past values of the return are already known. That is why the volatility is “conditionally non-stochastic” (Shumway and Stoffer, 2011, p. 288) and is called “time-varying volatility” in some literature (Koopman et al., 2010).

SV Model

Next, we discuss the standard SV model. Suppose again that the returns y_t of an asset are given by (5.13). In the SV model it is assumed that the volatility is a stochastic process on its own and the squared log-volatility is considered as a latent variable following an autoregressive process. So, the volatility in a SV(p) model is defined as

$$\log\bigl(\sigma_t^2\bigr) = \phi_0 + \sum_{j=1}^{p} \phi_j \log\bigl(\sigma_{t-j}^2\bigr) + w_t, \tag{5.16}$$

where $w_t \sim \mathcal{N}(0, \sigma_w^2)$ is the noise of the squared log-volatility process. The SV model is usually specified with $h_t := \log\bigl(\sigma_t^2\bigr)$. In this case the volatility h_t is indeed stochastic as it has a stochastic noise term. Then, in the case of SV models, the basic SV(1) model is often considered

$$\begin{cases} y_t = e^{h_t/2}\varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0,1), \\ h_t = \phi_0 + \phi_1 h_{t-1} + w_t, & w_t \sim \mathcal{N}(0, \sigma_w^2). \end{cases} \tag{5.17}$$

Hautsch and Ou (2008a) provide a comprehensive list of different extensions of the standard stochastic volatility model. We will not discuss those models in detail and refer to the authors for a more detailed discussion. Some interesting extensions include the *Stochastic Volatility with normally distributed Jumps* (SVJ) model and the *Asymmetric Stochastic Volatility* (ASV) model, which tries to model the asymmetry that sudden negative price changes have more (negative) impact on a stock price compared with positive price changes.

Comparison

Gerlach and Tuyl (2006) compare the GARCH$(1, 1)$ model, the SV(1) model and some extensions of both with each other on daily CAD/USD exchange rate data and S&P500 daily return index data. The authors find that in general the SV models outperform the GARCH models, but the best model in their research is a GARCH$(1, 1)$ with a Student's-t distributed noise (t-GARCH). Moreover, S. Kim and Shephard (1998) also compare the standard SV model with a standard GARCH and t-GARCH model, but for a JPY/USD exchange rate. They find that the t-GARCH and SV perform arguably comparable, but both models fit the data better than the standard GARCH model. So, at least for exchange rate data, the SV model seems to model the data better than the standard GARCH model. Nevertheless, a t-GARCH model could be an interesting extension as well. Preminger and Hafner (2010) argue that the SV model seems to provide better fit to data as it has two noise terms. In general, the volatility in the SV model requires an extra stochastic process to be modeled, whereas the volatility in the GARCH model is driven by past observations (y_{t-1}^2) and recursively on itself (σ_{t-1}^2). So, a GARCH model requires generally less computational effort than a SV model. According to Hautsch and Ou (2008a) another difficulty of SV models is that the likelihood cannot be derived into some closed-form expression. This is in line with Preminger and Hafner (2010) that state that GARCH models are preferred in practical situations.

All in all, the literature seems inconclusive on what model to use for volatility modeling in practice, so it seems dependent on what is considered more important. Hence, a SV model seems to be more flexible in the sense that it allows for more complex extensions, but at cost of tractability and computational effort. Meanwhile, a standard GARCH model seems to be more practical and computationally more efficient.

5.2.2 Extensions of DNS with Volatility

In the literature researchers have incorporated volatility in yield curve models based on the original DNS model in various ways. One of the more popular models is the DNS with GARCH observation or GARCH state noise (Koopman et al., 2010). Another DNS model with volatility is a three-layer hierarchical model with SV in the state noise (Hautsch and Ou, 2008b). We will first discuss the model with GARCH observation noise by Koopman et al. (2010), as we will use a slightly modified version of this model (see Subsection 6.3.1). On the contrary, we will not elaborate on the GARCH state noise extension as the idea is very similar to the GARCH observation noise extension and we refer to Harvey et al. (1992) for a more detailed discussion on both type of extensions. Moreover, we will also discuss the model introduced by Hautsch and Ou (2008b). This way we can present the typical ways of adding volatility, through the observation noise or the state noise and through a GARCH or SV process. Finally, we provide some advantages and drawbacks of the discussed approaches.

DNS with GARCH Observation Noise

The DNS model with GARCH observation noise is the model that Koopman et al. (2010) introduce as *DNS-GARCH*. The extension proposed by the authors is based on an approach originally introduced by Harvey et al. (1992). In this approach the observation noise $\boldsymbol{\varepsilon}_t$ is decomposed into a one-dimensional noise term ε_t^*, through which GARCH effects are introduced, and a Gaussian white noise term $\boldsymbol{\varepsilon}_t^+$ as

$$\boldsymbol{\varepsilon}_t = \Gamma \varepsilon_t^* + \boldsymbol{\varepsilon}_t^+, \tag{5.18}$$

where $\Gamma \in \mathbb{R}^M$ is the so-called *volatility loading* that determines how much of the GARCH effects translate to the volatility of the yields with different maturities, $\varepsilon_t^* \sim \mathcal{N}(0, h_t)$ and $\boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+)$. The one-dimensional noise term is also referred to as a "common shock" process since it is the term that introduces some underlying volatility process for all maturities. Here, the variance of the noise term ε_t^*, denoted by h_t, is the volatility that is modeled as a GARCH$(1,1)$ process

$$h_t = \gamma_0 + \gamma_1 (\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1}, \tag{5.19}$$

where $\gamma_0, \gamma_1, \gamma_2 > 0$ to ensure positive h_t, $\gamma_1 + \gamma_2 < 1$ and $h_0 = \gamma_0(1 - \gamma_1 - \gamma_2)^{-1}$. Then, the resulting model is given by

$$\begin{cases} \boldsymbol{y}_t &= \Lambda \boldsymbol{\beta}_t + \Gamma \varepsilon_t^* + \boldsymbol{\varepsilon}_t^+, \ \varepsilon_t^* \sim \mathcal{N}(0, h_t) \text{ and } \boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+), \\ \boldsymbol{\beta}_t &= (I - \Phi)\boldsymbol{\mu} + \Phi \boldsymbol{\beta}_{t-1} + \eta_t, \ \eta_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta), \\ h_t &= \gamma_0 + \gamma_1 (\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1}. \end{cases} \tag{5.20}$$

Notice that the volatility h_t is now dependent on the past observation noise term ε^*_{t-1}. This noise term is not directly observable, so we cannot use data as would be the case with past observations Y_{t-1}. The authors use the suggestion of Harvey et al. (1992) to use the expected value of $(\varepsilon^*_{t-1})^2$ given past observations Y_{t-1}. So, the volatility can be approximated as

$$h_t = \gamma_0 + \gamma_1 \mathbb{E}[(\varepsilon^*_{t-1})^2 | Y_{t-1}] + \gamma_2 h_{t-1}, \tag{5.21}$$

for which Harvey et al. (1992) note that this conditional expectation can be written as

$$\mathbb{E}[(\varepsilon^*_{t-1})^2 | Y_{t-1}] = \mathbb{E}[\varepsilon^*_{t-1} | Y_{t-1}]^2 + \mathrm{Var}[\varepsilon^*_{t-1} | Y_{t-1}], \tag{5.22}$$

which are exactly the conditional expectation and variance that are estimated by the Kalman filter. Hence, if the scalar noise term ε^*_t is augmented to the state vector as $(\boldsymbol{\beta}_t, \varepsilon^*_t)$, then the Kalman filter can be modified such that h_t is approximated for each iteration by

$$h_t \approx \gamma_0 + \gamma_1 \left[((\hat{\varepsilon}^*)^{t-1}_{t-1})^2 + (p_\varepsilon)^{t-1}_{t-1} \right] + \gamma_2 h_{t-1}, \tag{5.23}$$

where $(\hat{\varepsilon}^*)^{t-1}_{t-1}$ is the state estimate from the filtering step of the Kalman filter and $(p_\varepsilon)^{t-1}_{t-1}$ is the variance estimate for the specific state from the filtering step of the Kalman filter, which is equal to the last diagonal entry of the covariance matrix P^{t-1}_{t-1}. This results in the following conditionally Gaussian state-space model.

$$\boldsymbol{y}_t = \begin{bmatrix} \Lambda & \Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon^*_t \end{bmatrix} + \boldsymbol{\varepsilon}^+_t, \ \boldsymbol{\varepsilon}^+_t \sim \mathcal{N}(\mathbf{0}, \Sigma^+_\varepsilon), \tag{5.24}$$

$$\begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon^*_t \end{bmatrix} = \begin{bmatrix} (I-\Phi)\boldsymbol{\mu} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{t-1} \\ \varepsilon^*_{t-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon^*_t \end{bmatrix}, \ \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon^*_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_\eta & 0 \\ 0 & h_t \end{bmatrix} \right), \tag{5.25}$$

where the state is conditionally Gaussian given the past value of the now state variable ε^*_{t-1} and the variance h_{t-1} itself. Notice that the GARCH observation noise term ε^*_t is now both a state variable and a noise term, of which the original authors Harvey et al. (1992) also note that this is "somewhat unusual" (p. 131).

In summary, the DNS-GARCH model allows modeling volatility without a lot of additional computational burden. However, requiring that bond yields across all maturities are driven by the same volatility process is quite restrictive. Although the volatility loadings $\Gamma = [\Gamma_1, \ldots, \Gamma_{11}]^T$ try to model the different volatilities for the various maturities.

DNS with SV State Noise

The DNS model with SV state noise is the model that Hautsch and Ou (2008b) introduce as the *Stochastic Volatility Nelson-Siegel* (SVNS) model, for which they also introduce a MCMC algorithm for state and parameter estimation (Hautsch and Yang, 2012). Consider the modified DNS model

$$\begin{cases} \boldsymbol{y}_t = \Lambda \boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon), \\ \boldsymbol{\beta}_t = \boldsymbol{\mu} + \Phi \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta), \end{cases} \tag{5.26}$$

where $\boldsymbol{\beta}_{i,t}$ for $i = 1, 2, 3$ are uncorrelated and follow their distinct AR(1) process. Notice that the authors define the AR processes without a mean-reversion term $(I - \Phi)\boldsymbol{\mu}$, but just a standard intercept term $\boldsymbol{\mu}$ as opposed to the original authors Diebold et al. (2006). Then, Hautsch and Ou (2008b) add the volatility process via the state noise Σ_η by assuming the state noise is time-varying $\Sigma_\eta = \Sigma_{\eta,t} = \text{diag}(h_t^1, h_t^2, h_t^3)$ and that it follows a SV(1) process given by

$$\begin{bmatrix} \log h_t^1 \\ \log h_t^2 \\ \log h_t^3 \end{bmatrix} = \begin{bmatrix} \mu_h^1 \\ \mu_h^2 \\ \mu_h^3 \end{bmatrix} + \begin{bmatrix} \phi_h^1 & 0 & 0 \\ 0 & \phi_h^2 & 0 \\ 0 & 0 & \phi_h^3 \end{bmatrix} \begin{bmatrix} \log h_{t-1}^1 \\ \log h_{t-1}^2 \\ \log h_{t-1}^3 \end{bmatrix} + \begin{bmatrix} \xi_t^1 \\ \xi_t^2 \\ \xi_t^3 \end{bmatrix}, \tag{5.27}$$

$$\Leftrightarrow \text{diag}(\log \Sigma_{\eta,t}) = \boldsymbol{\mu}_h + \Phi_h \text{diag}(\log \Sigma_{\eta,t-1}) + \boldsymbol{\xi}_t, \tag{5.28}$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_h)$, $\Sigma_h = \text{diag}((\sigma_h^1)^2, (\sigma_h^2)^2, (\sigma_h^3)^2)$ and h_t^i for $i = 1, 2, 3$ are called the *factor volatilities*. An advantage of this model is that it allows for natural interpretation like the original DNS model of Diebold et al. (2006). The authors interpret h_t^1 as underlying macroeconomic volatility that touches the entire yield curve as it is the variance of the yield curve level. The second factor h_t^2 is the volatility in the yield curve slope, which can be interpreted as the volatility in yield spreads, given by $y_t(+\infty) - y_t(0)$. Finally, the third factor h_t^3 is associated with the volatility in bond yields with a medium-term maturity. It is also a straightforward approach, which follows the idea similar to the approach of Diebold et al. (2006) treating $\boldsymbol{\beta}_t$ as a latent variable. A drawback of this model is that we cannot use the theory of state-space models on this three-layer model directly. In order to estimate the volatility factors and the latent variables of this model the authors propose using a so-called Gibbs sampler and the Metropolis-Hastings algorithm. However, if we have T observations of bond yields for M different maturities and six latent variables, then we have to estimate $6T$ latent variables and $19 + M$ parameters. One can imagine that this becomes very difficult to employ in practice. So, even though a three-layer model allows for a natural way to add volatility in the DNS model, it also requires a lot of computational effort to use. Perhaps a modified model with SV volatility through the observation noise could balance performance and computational efficiency as this would result into a state-space model, for which we can use particle filters. However, we have not encountered this variant in the literature.

5.2.3 Concluding Remarks

In this section we reviewed literature on modeling yield curve volatility to find an appropriate volatility extension for a DNS model. We have found that a GARCH process and a Stochastic Volatility process are the most common ways to model volatility. Additionally, such a volatility process can be added through the observation noise or through the state noise in state-space models. The literature seems inconclusive to what model is preferred in the context of interest rates. However, a big advantage of the volatility extension with a GARCH process of Koopman et al. (2010) is that it lets us use a modified Kalman filter that does not require much computational effort. In contrast, we have to resort to more computationally heavy algorithms like Sequential Monte Carlo when using a SV process to model volatility as in the model of Hautsch and Ou (2008b). However, it seems that a SV(1) process outperforms a standard GARCH(1,1) process in modeling volatility for at least exchange rate data. Additionally, a SV extension in the observation noise could be an interesting volatility extension as it could balance performance and required computational power.

In summary, a GARCH extension as proposed by Koopman et al. (2010) in either the observation or state noise seems the most practical extension that shows relatively good performance, whereas a SV extension in the state noise as proposed by Hautsch and Ou (2008b) shows good performance, but a high computational cost. Moreover, a SV extension in the observation noise could be an interesting extension balancing computational effort and performance due to the aforementioned reasons, but we have not encountered such a volatility extension in the literature that could assert this claim.

6

Bayesian Yield Curve Modeling

In this chapter we first provide the general Bayesian setting for a state-space yield curve model. In particular, we discuss the theoretical setting and the practical details of using the Random Walk Metropolis algorithm. In the subsequent section we discuss the linear state-space yield curve models that are explored. Next, we also discuss the explored nonlinear yield curve models. Additionally, we will discuss the prior choice for the parameters for each model. Finally, in the last section we will derive the posterior predictive distribution for the models and provide a simulation algorithm based on that derivation for one-step ahead and h-step ahead forecasts.

For the readability, it is important to note that we have worked in a *modeling cycle*, where the various extended models are based on the results of the baseline state-space model (benchmark DNS). So, in this chapter we will provide the discussion of the models conceptually and provide the necessary preliminary work that is needed to produce the actual results. This means that the analyses that have resulted into the extensions can be found in Chapter 7.

6.1 Bayesian Setting

In this section we elaborate on the Bayesian setting of estimating parameters for the different yield curve models. Suppose we have T yield observations for M maturities, denoted by $Y_T = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\} \in \mathbb{R}^{M \times T}$, where for each time point $t = 1, \ldots, T$

$$\boldsymbol{y}_t = (y_t(\tau_1), \ldots, y_t(\tau_M))^T. \tag{6.1}$$

Let $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_d\} \in \mathbb{R}^d$ denote the collection of parameters of a state-space yield curve model. Then, we are interested in the posterior distribution $p(\boldsymbol{\psi}|Y_T)$. From Bayes' theorem (4.5) we know that

$$\begin{aligned} p(\boldsymbol{\psi}|Y_T) &\propto L(Y_t|\boldsymbol{\psi})p(\boldsymbol{\psi}), & (6.2)\\ &= e^{\ell(Y_t|\boldsymbol{\psi})}p(\boldsymbol{\psi}) & (6.3) \end{aligned}$$

Notice that the log-likelihood is precisely the log-likelihood from (3.82) that is computed (approximated) by the (modified) Kalman filter. In this setting we can write the log-likelihood as

$$\ell(Y_t|\boldsymbol{\psi}) = -\frac{MT}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log|\Sigma_t| - \frac{1}{2}\sum_{t=1}^{T}\boldsymbol{\epsilon}_t^T\Sigma_t^{-1}\boldsymbol{\epsilon}_t, \tag{6.4}$$

where $\boldsymbol{\epsilon}_t$ denotes the innovation term and Σ_t denotes the variance-covariance matrix as in Theorem 3.10. In order to keep the computational effort low for the RWM algorithm we assume that the parameters are mutually independent, i.e. $\psi_i \perp \psi_j$ for $i \neq j$. This also enables us to specify a prior on each parameter separately. Essentially, the assumption of independent parameters also reflects our lack of knowledge about the dependence structure between the parameters. This means that the posterior can be simplified further to

$$\begin{aligned} p(\boldsymbol{\psi}|Y_T) &\propto e^{\ell(Y_t|\boldsymbol{\psi})}p(\boldsymbol{\psi}) && (6.5)\\ &= e^{\ell(Y_t|\boldsymbol{\psi})}\prod_{i=1}^{d}p(\psi_i). && (6.6)\end{aligned}$$

The expression of the posterior in (6.5) means that if we specify the priors of the parameters for a model, then together with the (modified) Kalman filter we have all ingredients to use the Random Walk Metropolis algorithm.

6.1.1 Employing the Random Walk Metropolis Algorithm

Suppose we have chosen the priors and define the target distribution as $P(\boldsymbol{\psi}) := p(\boldsymbol{\psi}|Y_T)$. Then, in order to estimate the posterior of the parameters $\boldsymbol{\psi}$ for a model we need to specify the input variables of the RWM algorithm. As we have seen in Example 4.5 a good starting point can save a lot of time and a right scale is important for the convergence of the chains. However, there is no straightforward method that guarantees the RWM algorithm to converge and depending on the order of a parameter the algorithm can be quite sensitive on changes in scale $\boldsymbol{\sigma}$. That is why we will discuss how we choose the starting point $\boldsymbol{\psi}^{(0)}$ and adjust the scales $\sigma_1, \ldots, \sigma_d$ systematically. Moreover, we also discuss how we assess convergence.

Then, as discussed in Section 4.2.2, we will choose the maximum likelihood estimator (MLE) as the starting point, denoted by $\boldsymbol{\psi}^{(0)} := \boldsymbol{\psi}^{MLE}$. Since the MLE cannot be derived analytically, we have to use an optimization method to approximate the MLE. The used optimization method is a minimizer called *L-BFGS-B*, which is a so-called quasi-Newton optimization method[1]. The L-BFGS-B minimizer searches for the optimal set of parameters based on a projected gradient. Consequently, if we use non-informative priors, then the MLE will provide us with a starting point in a high density posterior region. In addition, the L-BFGS-B minimizer requires some initial values and search bounds for each parameter as well. So, we will provide both when presenting the results of the parameter estimation.

Moreover, as aforementioned there is no straightforward way of obtaining scales $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_d\}$ that result into the converging RWM chains right away. That is why we use a trial-and-error based approach, where we start with very small scales and adjust the scale per parameter. Whether a

[1] https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html

scale has to be changed is based on whether we see a white noise pattern visually for that particular parameter, indicating convergence. The trace plot for each parameter will be the main visual aid in assessing convergence together with the running mean of the chains. If a run seems successful visually we test for "actual" convergence with the Geweke diagnostic test (Theorem 4.7).

6.2 Linear Yield Curve Models

In this section we present the explored models that are variations of the DNS model as in (5.12) or are linear extensions of the DNS model. In this section we also elaborate on the current model that is used by the DSTA and the method that is currently employed to estimate the latent variables and parameters.

6.2.1 Current Model and Method

The DSTA uses the Nelson-Siegel model as in Diebold and Li (2006) and their two-step yield curve estimation approach. We note that this is not the only model used by the DSTA to calculate interest rate scenario's, but this model is used for calculating the interest rate costs. Consequently, we will refer to this model and method as the *current model and method.* Then, the used model is essentially the original Nelson-Siegel model for the yields $y_t(\tau)$ as in (5.1), but instead of treating the latent variables as parameters it is assumed that $\beta_{i,t},\ i = 1, 2, 3$ follow their distinct AR(1) process.

Suppose there are T yield observations for M maturities at each time t. The first step of the two-step estimation approach fixes λ at some value and estimates the values of $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ with the *ordinary least squares* (OLS) method for each separate time point t by

$$\hat{\boldsymbol{\beta}}_t = (\Lambda^T\Lambda)^{-1}\Lambda^T\boldsymbol{y}_t, \tag{6.7}$$

where $\Lambda \in \mathbb{R}^{M\times 3}$ is the observation matrix as in (5.12). Then, the λ value that minimizes the *residual sum of squares* (RSS) can be found by solving

$$\lambda^{opt} = \min_{\lambda\in D}\left\|\boldsymbol{y}_t - \Lambda\hat{\boldsymbol{\beta}}_t\right\|_2^2, \tag{6.8}$$

where $D \in \mathbb{R}$ is the search region for λ. Subsequently, the second step is based on using the fixed value λ^{opt} to estimate the AR(1) parameters for each $\beta_{i,t},\ i = 1, 2, 3$. So, by fixing λ^{opt} the latent variables can be estimated with the OLS method as in (6.7), which results in a sequence of estimated latent variables $(\hat{\boldsymbol{\beta}}_T) = \{\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_T\}$. Then, using the sequence of estimates $(\hat{\boldsymbol{\beta}}_T)$ the AR(1) parameters μ_i, ϕ_i in

$$\hat{\beta}_{i,t} = \mu_i + \phi_1\hat{\beta}_{i,t-1} + \eta_{i,t} \tag{6.9}$$

are estimated with the conditional maximum likelihood method and $\eta_{i,t}$ is modeled as a realization of a normal distribution, which has zero mean and has a variance that is equal to the variance of the residuals of the AR(1) fit. Finally, the latent variables are simulated by using the estimated parameters $\hat{\mu}_i, \hat{\phi}_i, \hat{\eta}_{i,t}$ and the bond yields can be computed by using the estimated latent variables and the Nelson-Siegel model as in (5.1). Using the last estimated latent variables $\hat{\boldsymbol{\beta}}_t$ as starting point an h-step ahead forecast is made by simulating the latent variables S number of times, which is shown in Figure 6.1.

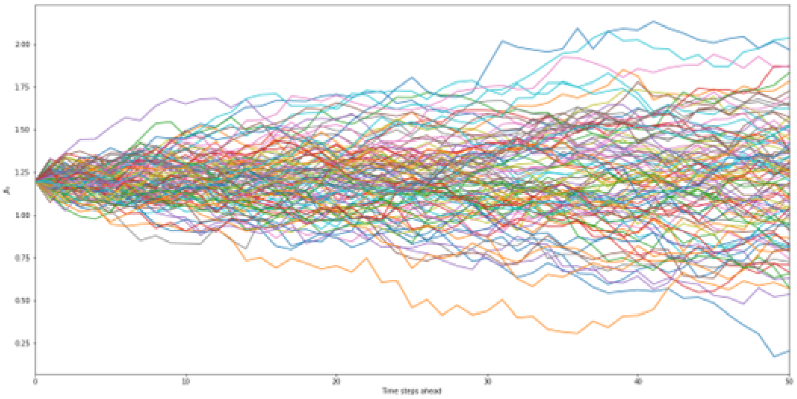


Figure 6.1: An example of $S = 100$ simulations of $\beta_{1,t}$, that goes 50 steps (months) ahead (*Source: DSTA*).

Consequently, the average of the S simulated paths of the latent variables at each time t is used as the forecast and the 5% of the top values of the simulations is used as a "worst-case" scenario forecast. An example of an actually used forecast is shown in Figure 6.2.

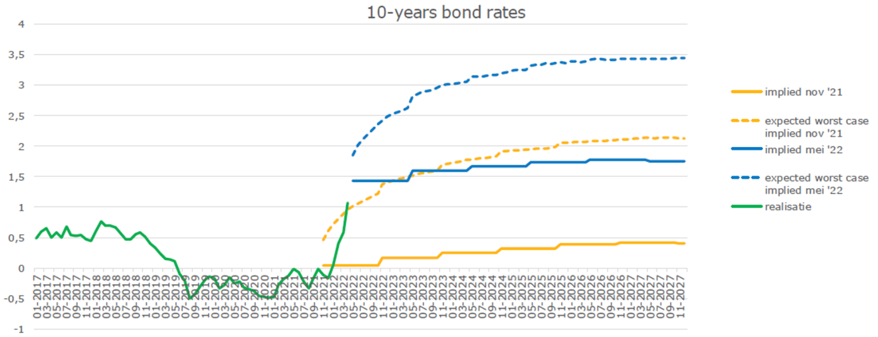


Figure 6.2: An example of a forecast for the Dutch 10 years (maturity $\tau = 360$ months) bond yield with the average forecasts (solid lines) and the worst-case scenario's (dashed lines) performed in November 2021 (yellow) and May 2022 (blue) and compared with the observed yields (solid green line) (*Source: DSTA*).

6.2.2 Benchmark DNS Model

All of the explored models are in the state-space framework. This allows us to use the theory of state-space models in Chapter 3, but complicates comparing the performance of these models (likelihoods, BIC values) and the methods (Bayesian approach, one-step filtering) with the current model and methods as described in the previous section. That is why we specify the *benchmark DNS* as starting point of modeling the bond yields with a state-space model to have some *benchmark* for further extensions within the state-space framework.

The benchmark DNS model is essentially a simplified version of the general DNS model in (5.12). For this model we assume that λ is a parameter that is not dependent on time t and that the bond yields $y_t(\tau_i)$ for every maturity τ_i have the same uncorrelated observation noise variance. In other words, we assume that the noise of the market observations is within the same bandwidth for yields across all maturities. This is a more restrictive assumption than Diebold et al. (2006) propose, but in the same line of thought. Then, the observations are modeled as

$$\boldsymbol{y}_t = \Lambda\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t,\ \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon), \tag{6.10}$$

where $\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t \in \mathbb{R}^{11}$, $\boldsymbol{\beta}_t \in \mathbb{R}^3$. Additionally, $\Lambda \in \mathbb{R}^{11\times 3}$ and $\Sigma_\varepsilon \in \mathbb{R}^{11\times 11}$ are given by

$$\Lambda = \begin{bmatrix} 1 & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} - e^{-\lambda\tau_1} \\ \vdots & \vdots & \vdots \\ 1 & \frac{1-e^{-\lambda\tau_{11}}}{\lambda\tau_{11}} & \frac{1-e^{-\lambda\tau_{11}}}{\lambda\tau_{11}} - e^{-\lambda\tau_{11}} \end{bmatrix},\ \Sigma_\varepsilon = \begin{bmatrix} \sigma^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2 \end{bmatrix}. \tag{6.11}$$

The latent variables $\boldsymbol{\beta}_t$ are assumed to follow their distinct AR(1) process. We assume the AR(1) process to be the same as the current model, which is a slightly modified version of the process Diebold et al. (2006) propose. The difference between the two formulations is that the general DNS model assumes some long-term mean reversion, whereas the current DSTA model assumes a standard AR(1) process. For this benchmark we also assume the noise variance of the latent variables to be equal, which is a very restrictive assumption. Intuitively, it is not obvious that the level, slope and curvature of the yield curve have a similar bandwidth. However, this is assumed for the sake of having a simple benchmark with not too much parameters. As a result, we model the latent variables as

$$\boldsymbol{\beta}_t = \boldsymbol{\mu} + \Phi\boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t,\ \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta), \tag{6.12}$$

where $\boldsymbol{\beta}_t, \boldsymbol{\eta}_t \in \mathbb{R}^3$, and $\boldsymbol{\mu} \in \mathbb{R}^3, \Phi, \Sigma_\eta \in \mathbb{R}^{3\times 3}$ are given by

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix},\ \Phi = \begin{bmatrix} \phi_1 & 0 & 0 \\ 0 & \phi_2 & 0 \\ 0 & 0 & \phi_3 \end{bmatrix},\ \Sigma_\eta = \begin{bmatrix} q^2 & 0 & 0 \\ 0 & q^2 & 0 \\ 0 & 0 & q^2 \end{bmatrix}. \tag{6.13}$$

So, the benchmark DNS state-space model is defined as

$$\begin{cases} \boldsymbol{y}_t = \Lambda\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(\sigma^2, \dots, \sigma^2)), \\ \boldsymbol{\beta}_t = \mu + \Phi\boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(q^2, q^2, q^2)), \end{cases} \tag{6.14}$$

with $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\eta}_t$ independent noise processes.

Prior Choice for the Parameters

The parameters of this model are $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma, q\}$. Notice that we should take extra care of the standard deviation parameters σ, q. An essential assumption for standard deviation or variance parameters is that they should be positive, but should preferably also be "invariant of rescaling" (Bailer-Jones, 2017, p. 115). This means that we choose the following improper prior

$$p(\sigma) \propto \frac{1}{\sigma}, \tag{6.15}$$

where $p(q)$ is defined analogously. Moreover, notice that a reparametrization with the log-transform results into

$$p(\log \sigma) \propto 1, \tag{6.16}$$

where the same holds for $p(\log q)$. Additionally, we usually require the AR(1) parameters to be $|\phi_i| < 1,\ i = 1, 2, 3$ in order to prevent explosive behaviour of the autoregressive processes. However, for the RWM algorithm we do not enforce a hard boundary on $\phi_i,\ i = 1, 2, 3$ by choosing, for instance, a uniform distribution $\mathcal{U}(-1, 1)$, because we want the chains to be able to explore the regions a bit above one or under minus one. After all, the likelihood term should result into $|\phi_i| < 1,\ i = 1, 2, 3$ if an explosive AR(1) is not a good model for the latent variables $\beta_{i,t},\ i = 1, 2, 3$. Finally, the parameters μ_1, μ_2, μ_3 are the intercept parameters of the latent variables, for which we do not specify any prior knowledge as well. We provide the choice of the priors in Table 6.3.

Table 6.3: The prior distribution for the parameters of the Benchmark DNS model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma$	$\log q$
Prior $p(\psi_i)$	$\mathbf{1}_{(0,+\infty)}$	1	1	1	1	1	1	1	1

6.2.3 DNS with Distinct State Noise

The DNS model with distinct state noise terms, or DNS-SN in short, is the same as the benchmark DNS model, but with distinct noise variance for each state instead of equal noise variance. This means that the yields and latent variables are modeled with the following state-space model

$$\boldsymbol{\beta}_t = \boldsymbol{\mu} + \Phi \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t,\ \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta), \tag{6.17}$$

where $\boldsymbol{\beta}_t, \boldsymbol{\eta}_t \in \mathbb{R}^3$ and $\boldsymbol{\mu} \in \mathbb{R}^3, \Phi, \Sigma_\eta \in \mathbb{R}^{3\times 3}$ are given by

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix},\ \Phi = \begin{bmatrix} \phi_1 & 0 & 0 \\ 0 & \phi_2 & 0 \\ 0 & 0 & \phi_3 \end{bmatrix},\ \Sigma_\eta = \begin{bmatrix} q_1^2 & 0 & 0 \\ 0 & q_2^2 & 0 \\ 0 & 0 & q_3^2 \end{bmatrix}. \tag{6.18}$$

So, the DNS-SN model is defined as the state-space model

$$\begin{cases} \boldsymbol{y}_t = \Lambda \boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(\sigma^2, \ldots, \sigma^2)), \\ \boldsymbol{\beta}_t = \boldsymbol{\mu} + \Phi \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(q_1^2, q_2^2, q_3^2)), \end{cases} \tag{6.19}$$

with $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\eta}_t$ again independent noise processes.

Prior Choice for the Parameters

The parameters of this model are $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma, q_1, q_2, q_3\}$. For this model the same holds again for the standard deviation parameters and the autoregression parameters. The choice of the prior for each parameter is provided in Table 6.4.

Table 6.4: The prior distribution for the parameters of the DNS-SN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log\sigma$	$\log q_1$	$\log q_2$	$\log q_3$
Prior $p(\psi_i)$	$\mathbf{1}_{(0,+\infty)}$	1	1	1	1	1	1	1	1	1	1

6.2.4 DNS with Autoregressive Observation Noise and Random Walk States

The DNS model with autoregressive observation noise and random walk states, or DNS-ARRW, is the benchmark DNS model with two extensions. The first extension is the assumption that the observation noise $\boldsymbol{\varepsilon}_t$ can be decomposed as

$$\boldsymbol{\varepsilon}_t = \boldsymbol{\varepsilon}'_t + \boldsymbol{\nu}_t, \tag{6.20}$$

$$\tag{6.21}$$

where $\boldsymbol{\varepsilon}'_t \in \mathbb{R}^{11}$ is a first order autoregressive component and $\boldsymbol{\nu}_t \in \mathbb{R}^{11}$ is a Gaussian white noise component. The autoregressive part $\boldsymbol{\varepsilon}'_t$ can be seen as some underlying process that models underlying market processes that have some memory or prolonged effect on the bond yield, whereas the white noise part $\boldsymbol{\nu}_t$ can be seen as the remaining noise. The process $\boldsymbol{\varepsilon}_t$ is modeled as

$$\boldsymbol{\varepsilon}'_t = A\boldsymbol{\varepsilon}'_{t-1} + \boldsymbol{\xi}_t,\ \boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\xi), \tag{6.22}$$

where $\boldsymbol{\varepsilon}'_t, \boldsymbol{\xi}_t \in \mathbb{R}^{11}$, and $A, \Sigma_\xi \in \mathbb{R}^{11\times 11}$ are defined as

$$A = \begin{bmatrix} \alpha_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha_{11} \end{bmatrix},\ \Sigma_\xi = \begin{bmatrix} \sigma_\xi^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_\xi^2 \end{bmatrix}. \tag{6.23}$$

The second extension is the assumption that the latent variables $\boldsymbol{\beta}_t$ are a random walk process instead of an AR(1) process. Essentially, this is an AR(1) process with autoregression parameters equal to one. So, we model $\boldsymbol{\beta}_t$ as

$$\boldsymbol{\beta}_t = \boldsymbol{\mu} + \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, \tag{6.24}$$

which means that we assume that the level, slope and curvature of the yield curve are mainly governed by their past value and some random noise.

Finally, bringing the two extension together we obtain the following model.

$$\begin{cases} \boldsymbol{y}_t = \Lambda \boldsymbol{\beta}_t + \boldsymbol{\varepsilon}'_t + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\nu), \\ \boldsymbol{\beta}_t = \boldsymbol{\mu} + \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta), \\ \boldsymbol{\varepsilon}'_t = A\boldsymbol{\varepsilon}'_{t-1} + \boldsymbol{\xi}_t, & \boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\xi), \end{cases} \tag{6.25}$$

where $\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t, \boldsymbol{\nu}_t, \boldsymbol{\xi}_t \in \mathbb{R}^{11}$, $\boldsymbol{\beta}_t, \boldsymbol{\eta}_t \in \mathbb{R}^3$ and $\Lambda \in \mathbb{R}^{11\times 3}$ is defined as in the previous models. Moreover, $A, \Sigma_\xi \in \mathbb{R}^{11\times 11}$ are defined as in (6.23) and Σ_ν is defined as

$$\Sigma_\nu = \begin{bmatrix} \sigma_\nu^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_\nu^2 \end{bmatrix}. \tag{6.26}$$

Rewriting this into a linear state-space model is straightforward and is given by

$$\begin{cases} \boldsymbol{y}_t = \begin{bmatrix} \Lambda & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_t \\ \boldsymbol{\varepsilon}_t \end{bmatrix} + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\nu), \\ \begin{bmatrix} \boldsymbol{\beta}_t \\ \boldsymbol{\varepsilon}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} I & O \\ O^T & A \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{t-1} \\ \boldsymbol{\varepsilon}_{t-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \boldsymbol{\xi}_t \end{bmatrix}, & \begin{bmatrix} \boldsymbol{\eta}_t \\ \boldsymbol{\xi}_t \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_\eta & O \\ O^T & \Sigma_\xi \end{bmatrix}\right), \end{cases} \tag{6.27}$$

where $O \in \mathbb{R}^{3\times 11}$ is the matrix with zero-only entries.

Recall that it follows from Corollary 3.8 that this state-space model is observable, which is already shown in Example 3.9. So, the DNS-ARRW model is a well-defined state-space model, for which we can correctly use the Kalman filter and related state-space theory.

Prior Choice for the Parameters

The parameters of this model are $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, q, \sigma_\nu, \alpha_1, \dots, \alpha_{11}, \sigma_\xi\}$. For this model the same holds again for the standard deviation parameters and the autoregression parameters. The choice of the prior for each parameter is provided in Table 6.5.

Table 6.5: The prior distribution for the parameters of the DNS-ARRW model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	$\log q$	$\log \sigma_\nu$	α_1	α_2	α_3
Prior $p(\psi_i)$	$\mathbf{1}_{(0,+\infty)}$	1	1	1	1	1	1	1	1
Parameter ψ_i	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	$\log \sigma_\xi$
Prior $p(\psi_i)$	1	1	1	1	1	1	1	1	1

6.3 Nonlinear Yield Curve Models

In this section we discuss the nonlinear extensions of the benchmark DNS model, which are the DNS with GARCH observation noise as introduced by Koopman et al. (2010) (see Subsection 5.2.2) and the DNS model with distinct observation and state noise and a GARCH observation noise, which is also based on Koopman et al. (2010).

6.3.1 DNS with GARCH Observation Volatility

We use the slightly modified version of the DNS-GARCH model of Koopman et al. (2010) without the mean-reversion term that has volatility modeled through the observation noise. The DNS model with GARCH Observation Volatility, or DNS-OV in short, is given by

$$\begin{cases} \boldsymbol{y}_t = \begin{bmatrix} \Lambda & \Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} + \boldsymbol{\varepsilon}_t^+, & \boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+), \\ \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{t-1} \\ \varepsilon_{t-1}^* \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix}, & \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_\eta & 0 \\ 0 & h_t \end{bmatrix}\right), \\ h_t = \gamma_0 + \gamma_1 (\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1}, \end{cases} \tag{6.28}$$

where $\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t^+ \in \mathbb{R}^{11}$, $\boldsymbol{\beta}_t, \boldsymbol{\eta}_t \in \mathbb{R}^3$, $\varepsilon_t^*, h_t, \gamma_0, \gamma_1, \gamma_2 \in \mathbb{R}$, $\Lambda \in \mathbb{R}^{11\times 3}, \Sigma_\eta, \Phi \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{\mu} \in \mathbb{R}^3$ are the same as for the benchmark DNS model. In addition, $\Gamma \in \mathbb{R}^{11}$ and $\Sigma_\varepsilon^+ \in \mathbb{R}^{11\times 11}$ are defined as

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_{11} \end{bmatrix}, \; \Sigma_\varepsilon^+ = \begin{bmatrix} (\sigma^+)^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (\sigma^+)^2 \end{bmatrix}. \tag{6.29}$$

In the concluding remarks of the literature review on modeling volatility we (Subsection 5.2.3) already argued the advantages and drawbacks of certain volatility extensions. The reason for choosing the GARCH model in the observation noise of Koopman et al. (2010) is twofold. First, modeling the volatility with a GARCH model results into a yield curve model that is still a state-space model, which means that we can use the modified Kalman filter to estimate states as described in Subsection 5.2.2. This reduces a lot of required computational effort compared with other filters like particle filters, which would mean that we have to employ two simulation algorithms (simulation for state estimation and RWM for parameter estimation). So, we can still perform parameter estimation in a practical amount of time. Secondly, Koopman et al. (2010) find that the extension with a GARCH process in the observation noise outperforms the one with a GARCH process in the state noise.

Prior Choice for the Parameters

The parameters of this model are $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma^+, q, \gamma_0, \gamma_1, \gamma_2, \Gamma_1, \dots, \Gamma_{11}\}$. Notice that $\gamma_0 = 0.0001$ is fixed and considered as a known constant for parameter identification purposes (Koopman et al., 2010, p. 332). For this model the same holds again for the standard deviation parameters and the autoregression parameters. Additionally, for the GARCH parameters γ_1, γ_2 the interval $(0, 1)$ is a hard constraint in the sense that it ensures that $h_0 \in \mathbb{R}$ and $h_0 > 0$. So, for the GARCH parameters γ_1, γ_2 we assume a uniform distribution on the interval $(0, 1)$. There could be a scenario that $\gamma_1 > 0.5$ and $\gamma_2 > 0.5$ resulting into $\gamma_1 + \gamma_2 > 1$. Regarding this scenario we can give a preview that in practice we see that as γ_1 increases, γ_2 decreases, preventing that the chains of both parameters result into $\gamma_1 + \gamma_2 > 1$. Furthermore, volatility is a process that is positive, since it is essentially a variance. So, we also assume that $\Gamma_i > 0,\ i = 1, \dots, 11$ to guarantee positive volatility processes for the yield of each maturity. Then, the choice of the priors are provided in Table 6.6.

Table 6.6: The prior distribution for the parameters of the DNS-OV model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma^+$
Prior $p(\psi_i)$	$\mathbf{1}_{(0,+\infty)}$	1	1	1	1	1	1	1
Parameter ψ_i	$\log q$	γ_1	γ_2					
Prior $p(\psi_i)$	1	$\mathcal{U}(0,1)$	$\mathcal{U}(0,1)$					

Moreover, we note that we originally used the RWM algorithm to approximate the posterior for the parameters $\Gamma_1, \ldots, \Gamma_{11}$ as well, with priors $p(\Gamma_i) \propto \mathbf{1}_{(0,+\infty)}$ for $i = 1, \ldots, 11$. However, since these parameters represent the proportion of volatility of a maturity relative to each other, the chains of the RWM runs show high mutual correlation. Consequently, the chains of $\Gamma_i,\ i = 1, \ldots, 11$ do not show convergence. Because of the difficult convergence and since the maximum likelihood estimates are similar to the results of Koopman et al. (2010), we fix the maximum likelihood estimates of $\Gamma_i,\ i = 1, \ldots, 11$. However, the uncertainty in the volatility process is still modeled as we still approximate the GARCH(1,1) parameters γ_1, γ_2. For further details we refer to Section 7.4.

6.3.2 DNS with GARCH Observation Volatility and Distinct Observation and State Noise

The DNS model with GARCH observation volatility and distinct observation and state noise, or DNS-OVOSN in short, is the model for which we have combined several findings from the results of the previous models. Essentially, the basis is the DNS-OV model, but with distinct state noise

$$\Sigma_\eta = \begin{bmatrix} q_1^2 & 0 & 0 \\ 0 & q_2^2 & 0 \\ 0 & 0 & q_3^2 \end{bmatrix}, \tag{6.30}$$

and observation noise for which the short (24 and 36 months), medium (48 to 108 months) and long-term (120, 240 and 360 months) maturities have the same variance. So, the observation noise covariance $\Sigma_\varepsilon^+ \in \mathbb{R}^{11\times 11}$ is given by

$$\Sigma_\varepsilon^+ = \begin{bmatrix} \Sigma_\varepsilon^S & & \emptyset \\ & \Sigma_\varepsilon^M & \\ \emptyset & & \Sigma_\varepsilon^L \end{bmatrix}, \tag{6.31}$$

where $\Sigma_\varepsilon^S = \text{diag}(\sigma_S^2, \sigma_S^2) \in \mathbb{R}^{2\times 2}, \Sigma_\varepsilon^M = \text{diag}(\sigma_M^2, \ldots, \sigma_M^2) \in \mathbb{R}^{6\times 6}$ and $\Sigma_\varepsilon^L = \text{diag}(\sigma_L^2, \sigma_L^2, \sigma_L^2) \in \mathbb{R}^{3\times 3}$. Additionally, the volatility loadings of different maturities are grouped together in the same way as the observation noise covariance. So, the volatility loading vector $\Gamma \in \mathbb{R}^{11}$ is defined as

$$\Gamma = [\Gamma_S, \Gamma_S, \Gamma_M, \Gamma_M, \Gamma_M, \Gamma_M, \Gamma_M, \Gamma_M, \Gamma_L, \Gamma_L, \Gamma_L]^T. \tag{6.32}$$

This means that the DNS-OVOSN model is defined as follows.

$$\begin{cases} \boldsymbol{y}_t = \begin{bmatrix} \Lambda & \Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} + \boldsymbol{\varepsilon}_t^+, & \boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+), \\ \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{t-1} \\ \varepsilon_{t-1}^* \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix}, & \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_\eta & 0 \\ 0 & h_t \end{bmatrix}\right), \\ h_t = \gamma_0 + \gamma_1 (\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1}, \end{cases} \tag{6.33}$$

where $\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t^+ \in \mathbb{R}^{11}$, $\boldsymbol{\beta}_t, \boldsymbol{\eta}_t \in \mathbb{R}^3$, $\varepsilon_t^*, h_t, \gamma_0, \gamma_1, \gamma_2 \in \mathbb{R}$, $\Lambda \in \mathbb{R}^{11\times 3}, \Phi \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{\mu} \in \mathbb{R}^3$ are the same as the benchmark DNS model, while $\Gamma \in \mathbb{R}^{11}, \Sigma_\eta \in \mathbb{R}^{3\times 3}, \Sigma_\varepsilon^+ \in \mathbb{R}^{11\times 11}$ are defined as aforementioned.

Prior Choice for the Parameters

The parameters of this model are $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma_S, \sigma_M, \sigma_L, q_1, q_2, q_3, \gamma_0, \gamma_1, \gamma_2, \Gamma_S, \Gamma_L, \Gamma_M\}$. Notice that we have fixed $\gamma_0 = 0.0001$ again. Subsequently, the choice of the priors for each parameter is the same as for the previous models. So, we have chosen priors that are uninformative on the specific domains of the parameters. In addition, we note that the volatility loadings $\Gamma_S, \Gamma_M, \Gamma_L$ are fixed again on the maximum likelihood estimates as was the case for the DNS-OV model due to the same reason of high mutual correlation resulting into bad convergence.

Table 6.7: The prior distribution for the parameters of the DNS-OVOSN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma_S$
Prior $p(\psi_i)$	$\mathbf{1}_{(0,+\infty)}$	1	1	1	1	1	1	1
Parameter ψ_i	$\log \sigma_M$	$\log \sigma_L$	$\log q_1$	$\log q_2$	$\log q_3$	γ_1	γ_2	
Prior $p(\psi_i)$	1	1	1	1	1	$\mathcal{U}(0,1)$	$\mathcal{U}(0,1)$	

6.4 Bayesian Forecasting

In this section we will derive the posterior predictive distribution for the one-step ahead future yield $\boldsymbol{y}_{t+1}$ given previous observations of bond yields Y_t. As we will see the posterior predictive distribution cannot be expressed in some closed-form. Therefore, we will also provide a simulation algorithm based on random sampling, which is a modified version of Algorithm 1. Moreover, we also derive the posterior predictive distribution for the multiple steps ahead yield $\boldsymbol{y}_{t+h}$ given the observations Y_t together with a simulation algorithm. Notice that a *step* in the context of the used data means a *month*, since we use monthly data.

6.4.1 Derivation of the Posterior Predictive Distribution

We are interested in the posterior predictive distribution $p(\boldsymbol{y}_{t+1}|Y_t)$. In order to derive the posterior predictive distribution we need the different (in)dependence relations of the models. Since all the models in Sections 6.2 and 6.3 are standard state-space model in the sense that they satisfy the independence relations (3.7) and (3.8), the analytical part holds for every used model. Then, we can derive the posterior predictive distribution as

$$
\begin{aligned}
p(\boldsymbol{y}_{t+1}|Y_t) &= \int\int p(\boldsymbol{y}_{t+1}, \boldsymbol{\psi}, \boldsymbol{\beta}_{t+1}|Y_t)\, d\boldsymbol{\psi}\, d\boldsymbol{\beta}_{t+1} && (6.34)\\
&\overset{(3.7)}{=} \int\int\int p(\boldsymbol{y}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_{t+1}, Y_t)p(\boldsymbol{\beta}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_t)p(\boldsymbol{\psi}, \boldsymbol{\beta}_t|Y_t)\, d\boldsymbol{\psi}\, d\boldsymbol{\beta}_{t+1}\, d\boldsymbol{\beta}_t && (6.35)\\
&\overset{(3.8)}{=} \int\int\int p(\boldsymbol{y}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_{t+1})p(\boldsymbol{\beta}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_t)p(\boldsymbol{\beta}_t|\boldsymbol{\psi}, Y_t)p(\boldsymbol{\psi}|Y_t)\, d\boldsymbol{\psi}\, d\boldsymbol{\beta}_{t+1}\, d\boldsymbol{\beta}_t. && (6.36)
\end{aligned}
$$

Notice that the expression of the posterior predictive distribution has become a three double integral with a product of four normal distributions. This cannot be expressed in some straightforward closed-form. This means that we have to resort to simulating the posterior predictive distribution.

6.4.2 One-Step Ahead Simulation

Simulating the posterior predictive distribution based on (6.36) is a less tedious task than one might expect from the integral form. It turns out that all components are already known or are already simulated for the posterior distribution. Notice that $p(\boldsymbol{\psi}|Y_t)$ is precisely the posterior distribution that we have already simulated with the Random Walk Metropolis algorithm. In addition, $p(\boldsymbol{\beta}_t|\boldsymbol{\psi}, Y_t)$ is the filter distribution that is estimated from the (modified) Kalman filter. Moreover, the state-transition distribution $p(\boldsymbol{\beta}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_t)$ and the distribution $p(\boldsymbol{y}_{t+1}|\boldsymbol{\psi}, \boldsymbol{\beta}_{t+1})$ are known as well, because these are the distribution that are defined by specifying the state-space model. Specifically, the state-transition and observation distributions are given by the state and observation equations respectively. From the known distributions we can derive a random sampling simulation algorithm, which is presented in Algorithm 3.

Algorithm 3: Simulation for the posterior predictive distribution of a yield curve model.

```
Input  : Posterior distribution $p(\boldsymbol{\psi}|Y_t)$ estimated by Algorithm 2.
Result: Posterior predictive samples $\boldsymbol{y}_{t+1}^{(1)}, \dots, \boldsymbol{y}_{t+1}^{(S)}$.
for $s = 1$ to $S$ do
    Sample $\boldsymbol{\psi}^{(s)} \sim p(\boldsymbol{\psi}|Y_t)$;
    Sample $\boldsymbol{\beta}_t^{(s)} \sim p(\boldsymbol{\beta}_t|\boldsymbol{\psi}^{(s)}, Y_t)$;
    Sample $\boldsymbol{\beta}_{t+1}^{(s)} \sim p(\boldsymbol{\beta}_{t+1}|\boldsymbol{\psi}^{(s)}, \boldsymbol{\beta}_t^{(s)})$;
    Sample $\boldsymbol{y}_{t+1}^{(s)} \sim p(\boldsymbol{y}_{t+1}|\boldsymbol{\psi}^{(s)}, \boldsymbol{\beta}_{t+1}^{(s)})$;
```

Using this algorithm we can compute a forecast estimator as $\hat{\boldsymbol{y}}_{t+1} = \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{y}_{t+1}^{(s)}$ and compute the associated credible regions as the q-th percentile of the simulated samples.

6.4.3 Multiple-Step Ahead Simulation

Besides one-step ahead forecasting we are also interested in forecasting yields that are multiple steps ahead. In particular, developments like interest rate hikes by central banks usually affect bond yields over a time period longer than one month. So, multiple step ahead forecasts are interesting for comparing the predicting power of the different models.

Subsequently, suppose that we want to forecast h steps ahead. Then, the posterior predictive distribution of $\boldsymbol{y}_{t+h}$ given the observations Y_t can be derived as

$$p(\boldsymbol{y}_{t+h}|Y_t) = \int \cdots \int p(\boldsymbol{y}_{t+h}, \ldots, \boldsymbol{y}_{t+1}|Y_t)\, d\boldsymbol{y}_{t+h-1}\ \ldots\ d\boldsymbol{y}_{t+1} \tag{6.37}$$

$$= \int \cdots \int p(\boldsymbol{y}_{t+1}|Y_t) \prod_{j=2}^{h} p(\boldsymbol{y}_{t+j}|\boldsymbol{y}_{t+j-1}, \ldots, \boldsymbol{y}_{t+1}, Y_t)\, d\boldsymbol{y}_{t+h-1}\ \ldots\ d\boldsymbol{y}_{t+1}, \tag{6.38}$$

where $p(\boldsymbol{y}_{t+1}|Y_t)$ is the posterior predictive distribution as in (6.36) and for $j = 2, \ldots, h$, the posterior predictive distribution $p(\boldsymbol{y}_{t+j}|\boldsymbol{y}_{t+j-1}, \ldots, \boldsymbol{y}_{t+1}, Y_t)$ is the same as in (6.36), but conditioned on $Y_t, \boldsymbol{y}_{t+1}, \ldots, \boldsymbol{y}_{t+j-1}$. Let $\hat{Y}_{t+j} = \{Y_t, \boldsymbol{y}_{t+1}, \ldots, \boldsymbol{y}_{t+j}\}$ denote the observations until time t and the future observations from time $t+1$ to $t+j$. In particular, we have

$$p(\boldsymbol{y}_{t+j}|\hat{Y}_{t+j-1}) \tag{6.39}$$

$$= \int \int \int p(\boldsymbol{y}_{t+j}|\boldsymbol{\psi}, \boldsymbol{\beta}_{t+j}) p(\boldsymbol{\beta}_{t+j}|\boldsymbol{\psi}, \boldsymbol{\beta}_{t+j-1}) p(\boldsymbol{\beta}_{t+j-1}|\boldsymbol{\psi}, \hat{Y}_{t+j-1}) p(\boldsymbol{\psi}|\hat{Y}_{t+j-1})\, d\boldsymbol{\psi}\ d\boldsymbol{\beta}_{t+j}\ d\boldsymbol{\beta}_{t+j-1}.$$

Then, based on (6.37) and (6.39) we can extend Algorithm 3 to a multiple step ahead simulation algorithm. Essentially, the idea of simulating multiple steps ahead instead of only one step is that we simulate S paths that are h steps long, for which each path naturally has the same set of parameters $\boldsymbol{\psi}^{(s)}$. In Algorithm 4 we present a simulation scheme to forecast multiple steps ahead. Notice that for $h = 1$ this is exactly Algorithm 3.

Algorithm 4: Simulation for the h-step ahead forecasts of a yield curve model.

```
Input  : Posterior distribution $p(\boldsymbol{\psi}|Y_t)$ estimated by Algorithm 2.
Result: $h$-step ahead paths $\{\boldsymbol{y}_{t+1}^{(1)}, \ldots, \boldsymbol{y}_{t+h}^{(1)}\}, \ldots, \{\boldsymbol{y}_{t+1}^{(S)}, \ldots, \boldsymbol{y}_{t+h}^{(S)}\}$.
for $s = 1$ to $S$ do
    Sample $\boldsymbol{\psi}^{(s)} \sim p(\boldsymbol{\psi}|Y_t)$;
    for $j = 0$ to $h - 1$ do
        Sample $\boldsymbol{\beta}_{t+j}^{(s)} \sim p(\boldsymbol{\beta}_{t+j}|\boldsymbol{\psi}^{(s)}, \hat{Y}_{t+j}^{(s)})$;
        Sample $\boldsymbol{\beta}_{t+j+1}^{(s)} \sim p(\boldsymbol{\beta}_{t+j+1}|\boldsymbol{\psi}^{(s)}, \boldsymbol{\beta}_{t+j}^{(s)})$;
        Sample $\boldsymbol{y}_{t+j+1}^{(s)} \sim p(\boldsymbol{y}_{t+j+1}|\boldsymbol{\psi}^{(s)}, \boldsymbol{\beta}_{t+j+1}^{(s)})$;
```

Then, using this algorithm we can compute a forecast estimator for future time $t+j$ as $\hat{\boldsymbol{y}}_{t+j} = \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{y}_{t+j}^{(s)}$ for $j = 1, \ldots, h$ and compute the associated credible regions as the q-th percentile of the simulated samples.

7

Results

In this chapter we will discuss the results for each model. As mentioned in Chapter 6, the results are grouped per model in chronological order. In particular, we provide three analyses per model. First, we discuss the estimated parameters obtained from the RWM algorithm. Secondly, we will discuss the estimated states obtained from the (modified) Kalman filter in the in-sample analysis sections. Here, in-sample means the “train data”. Finally, we will discuss the one-step ahead forecasts of each model. Then, in Section 7.6 we will compare the in-sample performance and the 12-months ahead forecasts of the current model and method with our explored models and method.

Before we dive into the actual results, we give some additional remarks on the in-sample analysis and forecasts. Recall that our motivation to model yields is driven by the fact that the current model and method are not able to realistically forecast a worst-case scenario of the yield curve, which is the upper bound of the 95% credible region, because of the volatile bond yields observed in the market. Specifically, past forecasts have been performed by the DSTA in May 2022 and in November 2021. Since the bond yields have increased quickly from November 2021 on, we will model the yield curves from March 2001 ($t = 0$) until November 2021 ($t = 249$) for the in-sample analyses and try to forecast from that point in time. Especially, because the yields across maturities have risen between two to four percentage points (190-380 bps) in the twelve months since November 2021 after a period of relatively stable and low yields. In addition, we will also provide a one-step ahead forecast for each model from August 2008 ($t = 90$) to September 2008 ($t = 91$), which has the largest absolute yield difference for a maturity (82 bps for the 24 months yield) and can be seen as a “worst-case” or “black swan” month.

7.1 Results of Benchmark DNS

In this section we go through the parameter, in-sample and forecasting results for the benchmark DNS model as specified in (6.14). Recall that the model is given by

$$\begin{cases} \boldsymbol{y}_t = \Lambda \boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\sigma^2, \dots, \sigma^2)), \\ \boldsymbol{\beta}_t = \mu + \Phi \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(q^2, q^2, q^2)), \end{cases} \tag{7.1}$$

with parameters $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma, q\}$.

7.1.1 Parameter Estimation

We denote the scales of the RWM algorithm as $\boldsymbol{\sigma}_{RWM}$ to avoid confusion. Then, the starting point and scales that result into converging chains are provided in Table 7.1 and 7.2.

Notice that for the benchmark DNS model we also have tried to estimate the Kalman filter initial state and initial covariance matrix $\hat{\boldsymbol{\beta}}^0 = [\hat{\beta}_1^0, \hat{\beta}_2^0, \hat{\beta}_3^0]^T$ and $P_0^0 = \mathrm{diag}(p^2, p^2, p^2)$. However, we could not find appropriate scales to find converging chains for these parameters. A reason for the difficult convergence could be that the observations give relatively strong new information as it seems that the observation noise is relatively small compared to the state noise. So, it could be that the initial values for the Kalman filter do not affect the log-likelihood strongly. That is why we will fix the initial values for the Kalman filter at the MLE values and try to find the posterior distribution of the model parameters $\boldsymbol{\psi}$ for this and the other models.

Table 7.1: The MLE values as RWM starting points $\boldsymbol{\psi}^{(0)}$ of the parameters approximated by the L-BFGS-B minimizer and the corresponding log-likelihood value for the benchmark DNS model.

Parameter ψ_i	λ	$\hat{\beta}_1^0$	$\hat{\beta}_2^0$	$\hat{\beta}_3^0$	p	μ_1	μ_2
Bounds	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$
Initial guess	0.1	0.0	0.0	0.0	1.0	0.0	0.0
MLE value	0.04771	0.00089	0.00016	-0.00012	0.99944	1×10^{-5}	-0.00044
Parameter ψ_i	μ_3	ϕ_1	ϕ_2	ϕ_3	σ	q	
Bounds	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$	
Initial guess	0.0	0.0	0.0	0.0	0.1	0.1	
MLE value	-0.00051	0.99301	0.97716	0.98113	0.0006	0.00312	
Log-likelihood	15333						

Table 7.2: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the benchmark DNS model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma$	$\log q$
Scale $\sigma_{i,RMW}$	0.0003	0.00016	0.00016	0.00016	0.004	0.0045	0.003	0.0062	0.013

Then, we run the RWM algorithm with the MLE values as starting points and the provided scales for 25000 iterations. The results are shown as trace plot for each parameter in Figure 7.3 and the estimated posterior distribution for each parameter is shown as a histogram in Figure 7.4. The average acceptance ratio of this run is $\bar{\alpha} \approx 0.3919$. This is higher than the 0.234 we aim for, but the average acceptance ratio is still reasonably between 0.1 and 0.4. Furthermore, the results of the Geweke diagnostic test with $\tau_A = 0.1, \tau_B = 0.5$ and significance level $\alpha = 0.05$ (Theorem 4.7) for the chains of each parameter are provided in Table 7.5. Recall that the Geweke diagnostic gives an indication whether the first and last segments have significantly different means. Consequently, the means of the first and last segments of the chains show no significant difference if $|G| < 1.96$. Then, the test results indicate that this is indeed the case. So, overall we assume that the chains have converged.

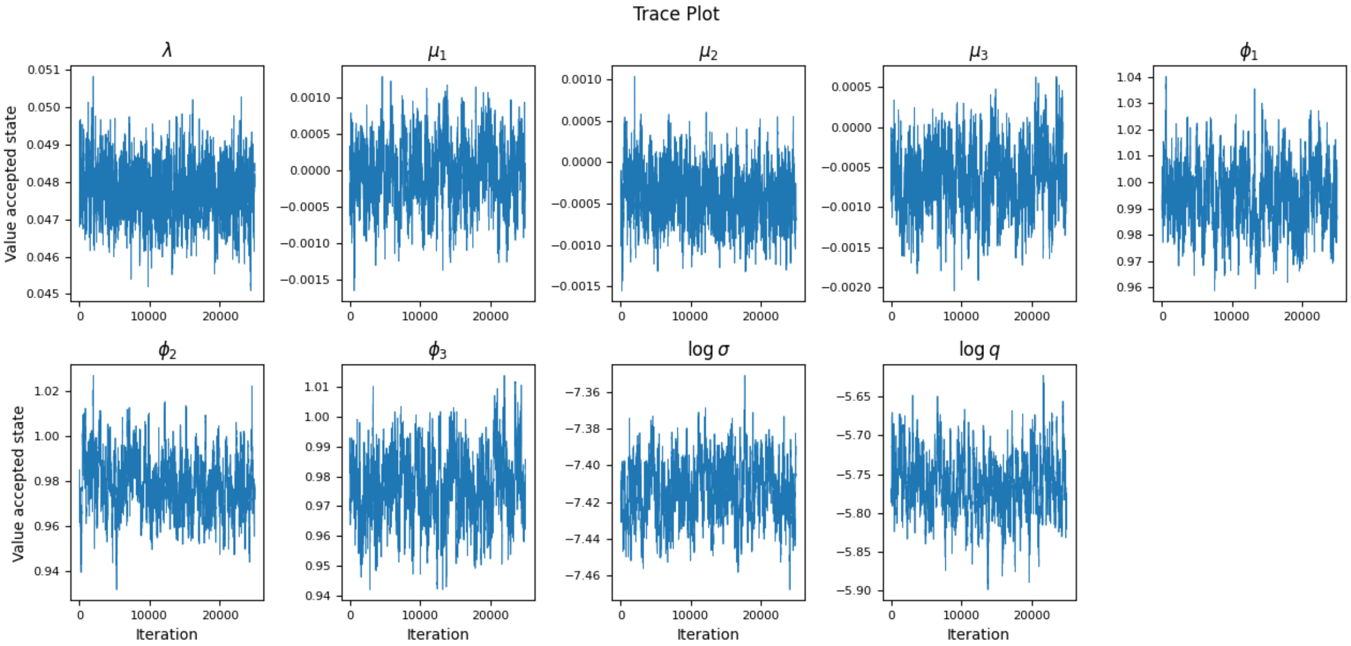


Figure 7.3: Trace plot of the chains of each parameter of the benchmark DNS model.

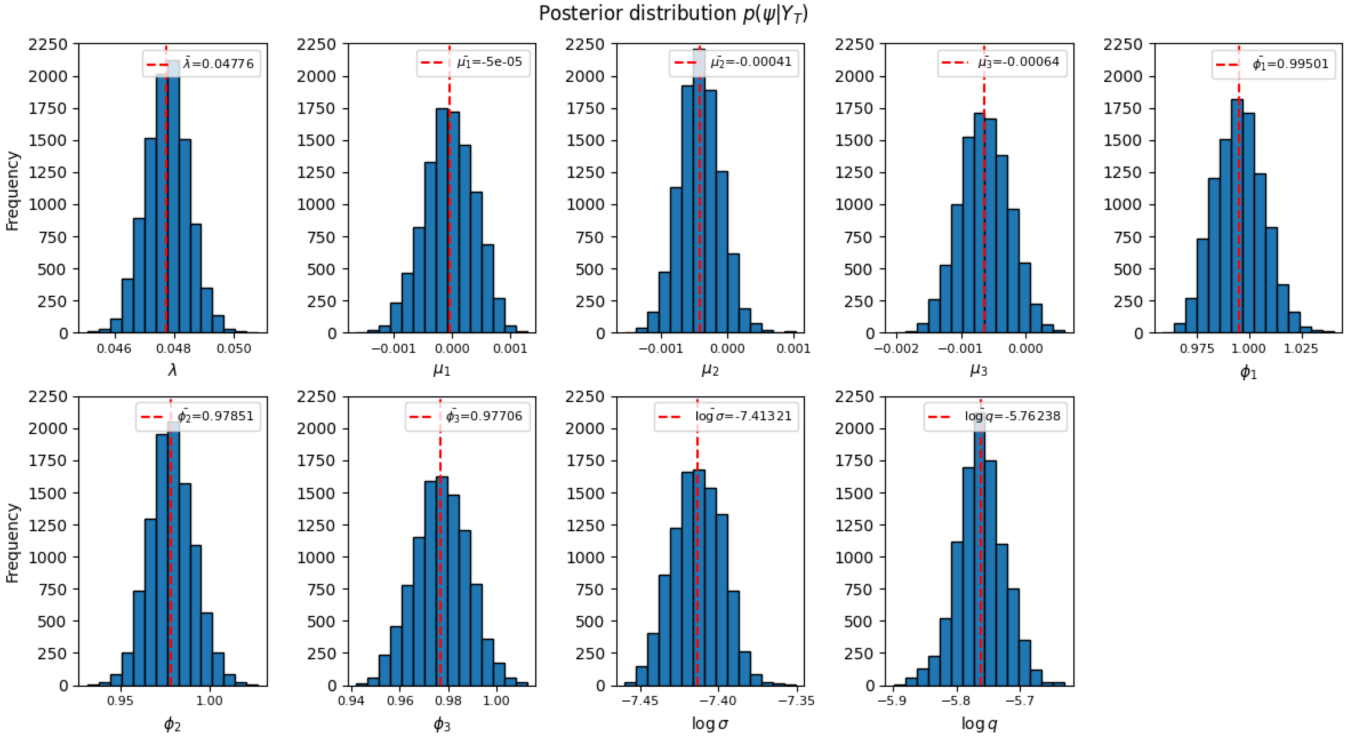


Figure 7.4: Histogram with the mean of the posterior distribution for each parameter of the benchmark DNS model.

Table 7.5: The results of the Geweke diagnostic test for the chains of each parameter of the benchmark DNS model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2
Statistic $\lvert G\rvert$	0.24665	0.16128	0.30633	0.20658	0.17749	0.38589
Means differ significantly?	No	No	No	No	No	No
Parameter ψ_i	ϕ_3	σ	q			
Statistic $\lvert G\rvert$ value	0.23123	0.23258	0.37681			
Means differ significantly?	No	No	No			

We use the *maximum a posteriori estimator* (MAPE), denoted by $\hat{\boldsymbol{\psi}}^{MAPE}$, for the further analyses. It turns out that the MLE is also the set of parameter values that yields the MAPE. Moreover, we notice that the parameters ϕ_1, ϕ_2, ϕ_2 are very close to one. This could indicate that the AR(1) processes of the state variables might have a so-called *unit root*. So, it is interesting to see whether $\phi_i = 1,\ i = 1, 2, 3$ would also provide a good fit. After all, if a model with $\phi_i = 1,\ i = 1, 2, 3$ can model the yields comparably well, then this reduces the amount of parameters by three, which is preferable for model extension that result in much more parameters.

In order to test for a unit root in the AR process of a time series, one can usually use a unit root test such as the Dickey-Fuller (DF) or augmented Dickey-Fuller (ADF) test (Shumway and Stoffer, 2011, Section 5.2). However, this requires us to have the actual observations of the state variables, which are not available per definition. Another option could be to use the estimated state variables as time series, on which a unit root test can be performed. However, in that case we have to specify the parameters before we can estimate the states. This would result into a situation where we test whether the state variables can be modeled by $\phi_i = 1,\ i = 1, 2, 3$ while using $\phi_i < 1,\ i = 1, 2, 3$ to estimate the state variables. So, one can imagine that the usual unit root testing does not allow us to perform a sound test. Then, another way to compare whether it makes sense to model the state variables as random walks instead of an AR(1) process is by fixing $\phi_i = 1,\ i = 1, 2, 3$, finding the MLE for this model and compare the log-likelihoods of the random walk and autoregressive model. Hence, we also provide the MLE of the model with unit roots, denoted by $\hat{\boldsymbol{\psi}}^{MLE}$. The parameter, log-likelihood and BIC values are shown in Table 7.6.

Table 7.6: The MAPE values $\hat{\boldsymbol{\psi}}^{MAPE}$ for the benchmark DNS model without unit roots and MLE values $\hat{\boldsymbol{\psi}}^{MLE}$ for the benchmark DNS model with unit roots and the corresponding log-likelihood (LL), BIC and AIC values.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2
MAPE $\hat{\psi}_i^{MAPE}$	0.04771	0.00001	-0.00044	-0.00051	0.99301	0.97716
MLE $\hat{\psi}_i^{MLE}$	0.04772	-0.00022	0.0	0.00005	(1.0)*	(1.0)*
Parameter ψ_i	ϕ_3	σ	q			
MAPE $\hat{\psi}_i^{MAPE}$	0.98113	0.0006	0.00312			
MLE $\hat{\psi}_i^{MLE}$	(1.0)*	0.0006	0.00312			
Model fit measure	LL	BIC	AIC			
No unit roots	**15333**	-30616	**-30648**			
Unit roots	15329	**-30625**	-30646			

* Fixed parameters.

We can see that the benchmark DNS model without unit roots ($\phi_i < 1,\ i = 1, 2, 3$) has a higher log-likelihood value, but the difference with the model that has unit roots ($\phi_i = 1,\ i = 1, 2, 3$) is relatively small. However, the additional benefit of fixing $\phi_i = 1,\ i = 1, 2, 3$ is the reduction in the amount of parameters. We see that the model with the unit roots has a quite lower BIC than the model without, which indicates that fixing $\phi_i = 1,\ i = 1, 2, 3$ sufficiently improves the model with less parameters. So, modeling the state variables as a random walk could be an interesting extension and we explore this in the DNS-ARRW model (Section 7.3).

7.1.2 In-Sample Analysis

In this section we discuss the estimated states and provide some yield curve estimations of various dates. Using the estimated states we can obtain the estimated yields for the observed maturities. We discuss the residuals of the resulting estimated yields as well.

Estimated State Variables and Yields

We use the Kalman filter with the MLE values of the initial state $\hat{\boldsymbol{\beta}}^0$ and initial standard deviation p as in Table 7.1 and the MAPE values for the model parameters as in Table 7.6. The estimated state variables $\hat{\boldsymbol{\beta}}_t$ until November 2021 are shown in Figure 7.7.

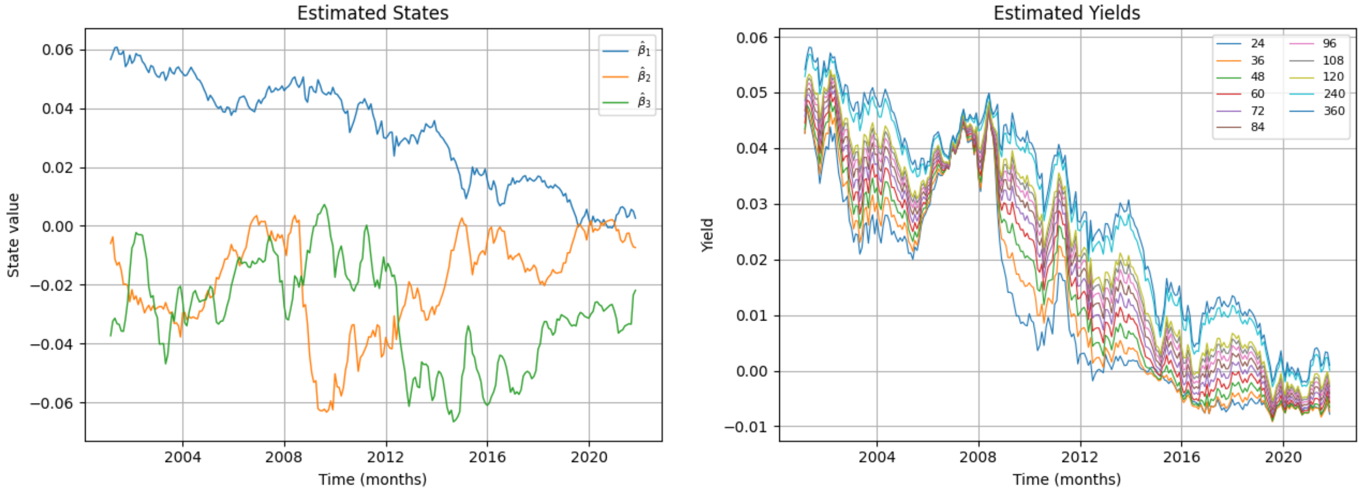


Figure 7.7: The estimated state variables $\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \hat{\beta}_{3,t}$ obtained by the Kalman filter and the estimated yields $\hat{\boldsymbol{y}}_t = \Lambda\hat{\boldsymbol{\beta}}_t$ for $t = 1, \ldots, 249$ with the benchmark DNS model.

We see that the level of the yield curves overall has decreased, which is as expected. More interestingly, we notice that during economic recessions (2007-2009, 2014-2015) or global turmoil like the covid-19 pandemic (2020-2021) the slope of the yield curves is around or slightly above zero. This means that the model can capture flat or slightly inverted yield curves when we expect the yield curve to be actually inverted. Moreover, we can see that the curvature of the yield curves until 2012 seems to be relatively less pronounced compared with the period after 2012. Perhaps this could be related with the start of the Quantitative Easening (QE) program of the ECB around 2015. Due to this program, the ECB started to buy various assets, of which government bonds, to decrease interest rates in an already low interest rate environment.

Residuals Analysis

We also discuss how well the benchmark DNS model fits the observed yields. To this end, the residuals as well as the ACF and PACF are analyzed. The residuals, ACF and PACF are shown in Figure 7.9, 7.10 and 7.11. Notice that the benchmark DNS model seems to approximate the yields quite well during periods of low volatility, whereas the residual is larger in periods of sudden increases or decreases of yields. The residuals, however, do not seem to resemble white noise as we have assumed. We perform the Ljung-Box test on the residuals to test whether the residuals are significantly a white noise process or not. In Table 7.8 the results of the Ljung-Box test with a significance level $\alpha = 0.05$ and lag $h = 1$ is shown. A lag of $h = 1$ means that serial correlation is tested between each residual at time t and $t - 1$. We can see that the p value for the residuals of every maturity is extremely low and far less than the significance level $\alpha = 0.05$, which indicates that the residuals are significantly not a white noise process.

Table 7.8: Results of the Ljung-Box test for serial correlation in the residuals of the estimated yields for each maturity with the benchmark DNS model.

Maturity τ_i	24	36	48	60	72	84
p value	7.1×10^{-27}	1.5×10^{-31}	1.4×10^{-29}	4.9×10^{-30}	8.0×10^{-19}	1.8×10^{-24}
White noise?	No	No	No	No	No	No
Maturity τ_i	96	108	120	240	360	
p value	1.7×10^{-32}	1.3×10^{-36}	4.2×10^{-36}	9.7×10^{-39}	1.3×10^{-42}	
White noise?	No	No	No	No	No	

The ACF and PACF strengthen the idea that the observation noise might not model the remaining effects that are not captured by the current state variables $\boldsymbol{\beta}_t$ well. An inspection of the ACF and PACF indicates that the observation noise shows some autoregressive behaviour. Moreover, notice that these effects could also stem from the state noise. However, since the state variables are not observable we cannot analyse the residual for the state variables. So, we only focus on the observation noise for the residuals analysis. In order to know what process would be appropriate to model the observation noise we compare the BIC values of different $ARMA(p, q)$ models for the residuals. It seems that the most significant orders are $p, q = 0, 1, 2$, so we compare all eight different combinations. The results are provided in Table 7.12. It seems that the most appropriate model for the observation noise is an AR(1) model. Together with the findings of the unit roots in the AR(1) process for the state variables these two results are explored in the DNS-ARRW model in Section 7.3.

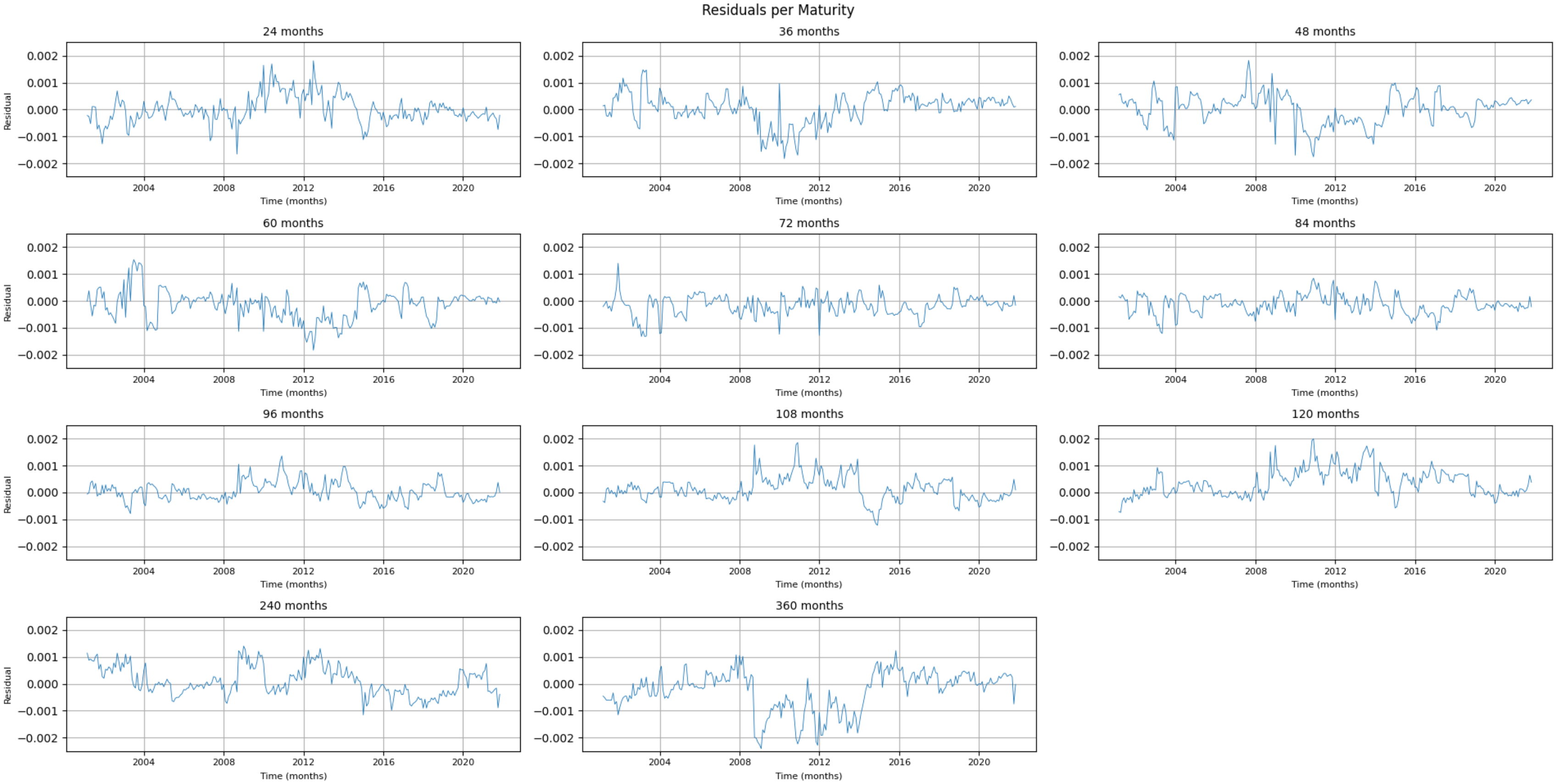


Figure 7.9: Residuals of the estimated yields for each maturity with the benchmark DNS model.

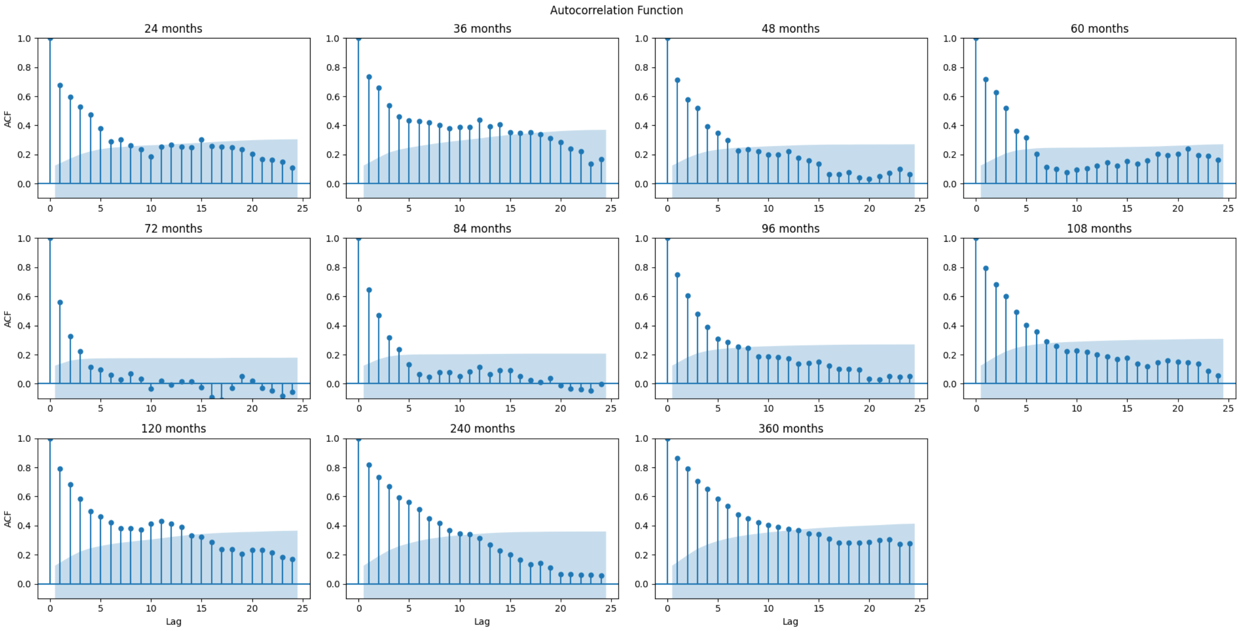


Figure 7.10: ACF of the residuals for each maturity with the benchmark DNS model.

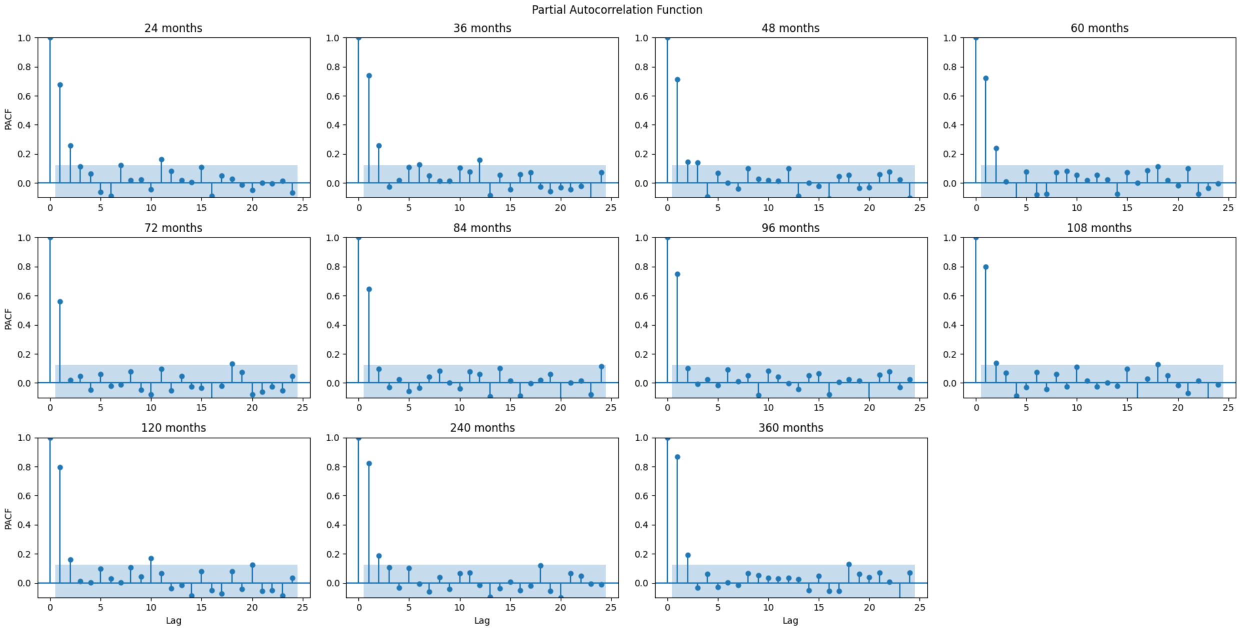


Figure 7.11: PACF of the residuals for each maturity with the benchmark DNS model.

Table 7.12: Results of the residuals fit of different ARMA models with order p, q.

(p, q)	(0,1)	(0,2)	**(1,0)**	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
AIC	-32721	-32801	**-36264**	-36096	-35968	-36214	-35980	-35793

7.1.3 Forecasting Analysis

In this section we present the results of the one-month ahead forecast of the yield curve in December 2021 ($t = 250$) and September 2008 ($t = 91$). The forecast is based on simulations of the posterior predictive distribution (Algorithm 3). For each maturity a posterior predictive sample is simultaneously simulated 1000 times. The forecast is shown in Figure 7.13 together with the observed yield curve of the forecast dates (December 2021 and September 2008) and the previous dates (November 2021 and August 2008). In both forecasts we also show the uncertainty that we would expect from only the observation noise. We provide the interval $\pm\sigma$, which shows the uncertainty that is within ± 1 standard deviation. We can see that the observed yield curve of December 2021 is entirely captured within the credible region of the forecast, whereas the observation noise would indicate that these values are very unlikely. However, the credible interval is still quite wide, as 95% of the are within one yield percentage point. For example, the bond yield for a maturity of 24 months in December 2021 is likely to be any value between -1.5% and -0.2%. A reason for the variability in the simulated yields could be that the state noise is two orders larger than the observation noise. This could result into a large deviation in the simulated state values $\hat{\boldsymbol{\beta}}_{T+1}$, which carries through to the yields. Recall that we assumed for the benchmark DNS model that the state noise of $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ are equal. However, looking at the interpretation of the state variables, it is quite restrictive to assume that the level, slope and curvature of the yield curve show equal variance. So, it could be that the state noise variance q^2 settles for some value that is too large for one state variable and too small for another. One way to reduce the variability in the state variable simulation could be to let each state variable have its distinct variance. We explore this further in Section 7.2 for the DNS-SN model.

Moreover, we notice that for the one-step ahead forecast of September 2008 the shortest-term maturity of 24 months is not within the 95% credible region, whereas the long-term maturities of 120, 240 and 360 are close to the mean values of the posterior predictive samples. However, we can see that the forecast of September 2008 is close to the yield curve of the prior month, which has seen the largest shift in the short-term maturities and to a lesser extent in the long-term maturities. Consequently, the forecast yields are for the most part inside the 95% credible region, whereas the shape of the yield curve is harder for the benchmark DNS model to forecast correctly when a large shock occurs.

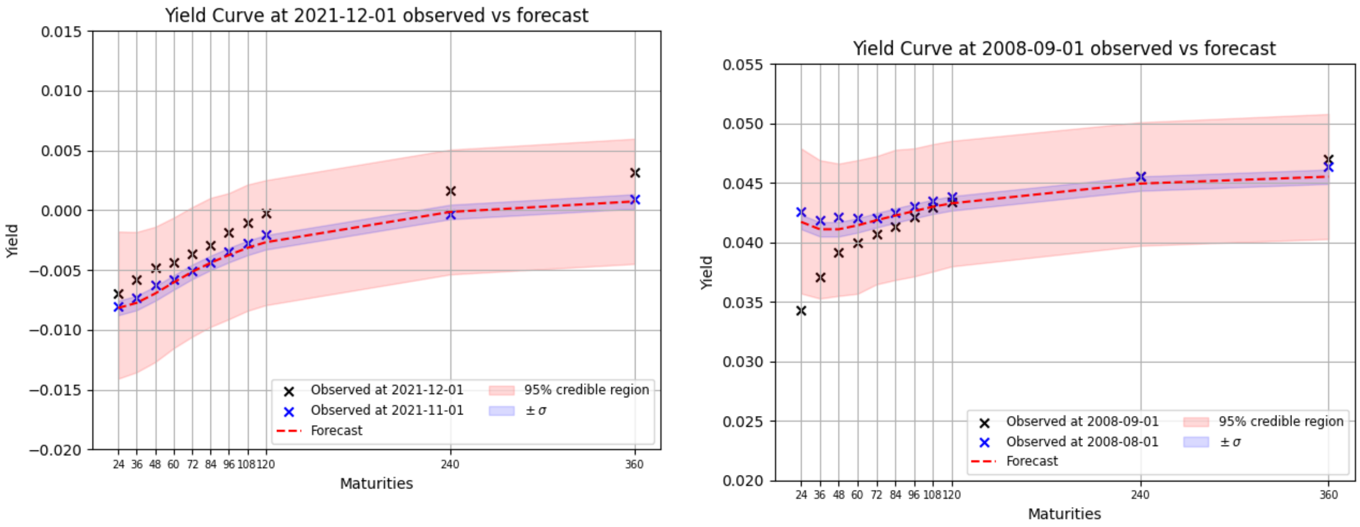


Figure 7.13: The one-month ahead forecast (dashed red line) of December 2021 (left) and September 2008 (right) with 95% credible regions (red surface) and with uncertainty due to the observation noise $\pm\sigma$ (blue surface).

7.2 Results of DNS-SN

In this section we discuss the parameter, in-sample and forecasting results for the DNS-SN model. Recall that this model has distinct state noise variances for each state as specified in (6.19) and is given by

$$\begin{cases} \boldsymbol{y}_t = \Lambda\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(\sigma^2, \dots, \sigma^2)), \\ \boldsymbol{\beta}_t = \boldsymbol{\mu} + \Phi\boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(q_1^2, q_2^2, q_3^2)), \end{cases} \tag{7.2}$$

with parameters $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma, q_1, q_2, q_3\}$. Notice that we do not model the state variables as random walks in this model, because we want to be able to check what affects the credible regions of the forecast. If we model the state variables as random walks and we model distinct state noise variance, then it is hard to assess what modification leads to a more accurate (or less accurate) forecast.

7.2.1 Parameter Estimation

The scales and starting point that result into convergence of the RWM algorithm are presented in Table 7.15 and 7.14.

Table 7.14: The MLE values as RWM starting points $\boldsymbol{\psi}^{(0)}$ of the parameters approximated by the L-BFGS-B minimizer and the corresponding log-likelihood value for the DNS-SN model.

Parameter ψ_i	λ	$\hat{\beta}_1^0$	$\hat{\beta}_2^0$	$\hat{\beta}_3^0$	p	μ_1
Bounds	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(-\infty,\infty)$
Initial guess	0.1	0.0	0.0	0.0	1.0	0.0
MLE value	0.04862	0.00272	0.00036	-0.0002	0.9989	-6×10^{-5}
Parameter ψ_i	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	σ
Bounds	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$
Initial guess	0.0	0.0	0.0	0.0	0.0	0.1
MLE value	-0.00044	-0.0011	0.99496	0.97722	0.96224	0.00059
Parameter ψ_i	q_1	q_2	q_3			
Bounds	$(10^{-4},\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$			
Initial guess	0.1	0.1	0.1			
MLE value	0.00203	0.0035	0.00501			
Log-likelihood	15392					

Table 7.15: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the DNS-SN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2
Scale $\sigma_{i,RMW}$	0.00045	0.0002	0.0002	0.00035	0.0077	0.0103
Parameter ψ_i	ϕ_3	$\log \sigma$	$\log q_1$	$\log q_2$	$\log q_3$	
Scale $\sigma_{i,RMW}$	0.013	0.009	0.026	0.0265	0.027	

Then, we run the RWM algorithm again with the MLE values as starting points and the presented scales for 20000 iterations. For each parameter the trace plot is shown in Figure 7.16 and the estimated posterior distribution for each parameter is presented as a histogram in Figure 7.17. The average acceptance ratio of this run is $\bar{\alpha} \approx 0.1339$. This indicates that the rate of convergence might be a bit slow, but it is still satisfactorily between 0.1 and 0.4 and it seems that the starting point is already in a high density region. Moreover, we use the Geweke test again with $\tau_A = 0.1, \tau_B = 0.5$ and significance level $\alpha = 0.05$, of which the results are shown in Table 7.18. We can see that the test indicates no significant difference between the means of the first and last segments for every parameter. All in all, we assume the chains to have converged. Subsequently, we use the *maximum a posteriori estimator* (MAPE) again for the in-sample analysis. For the DNS-SN model the MAPE is equal to the MLE as well and is provided in Table 7.19.

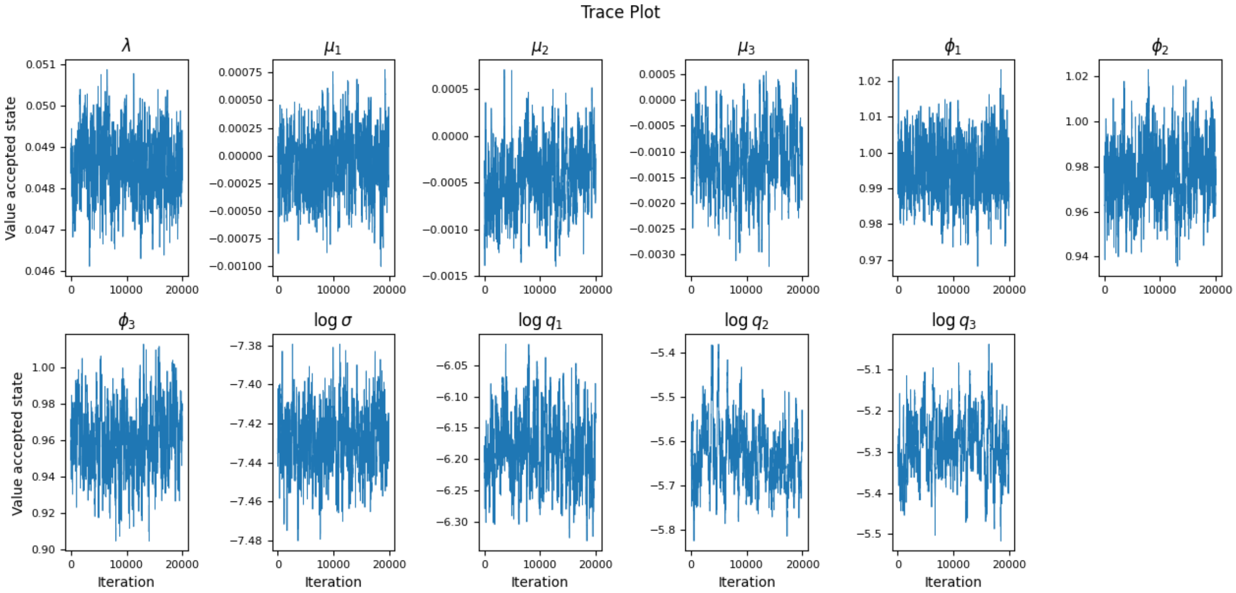


Figure 7.16: Trace plot of the chains for each parameter of the DNS-SN model.

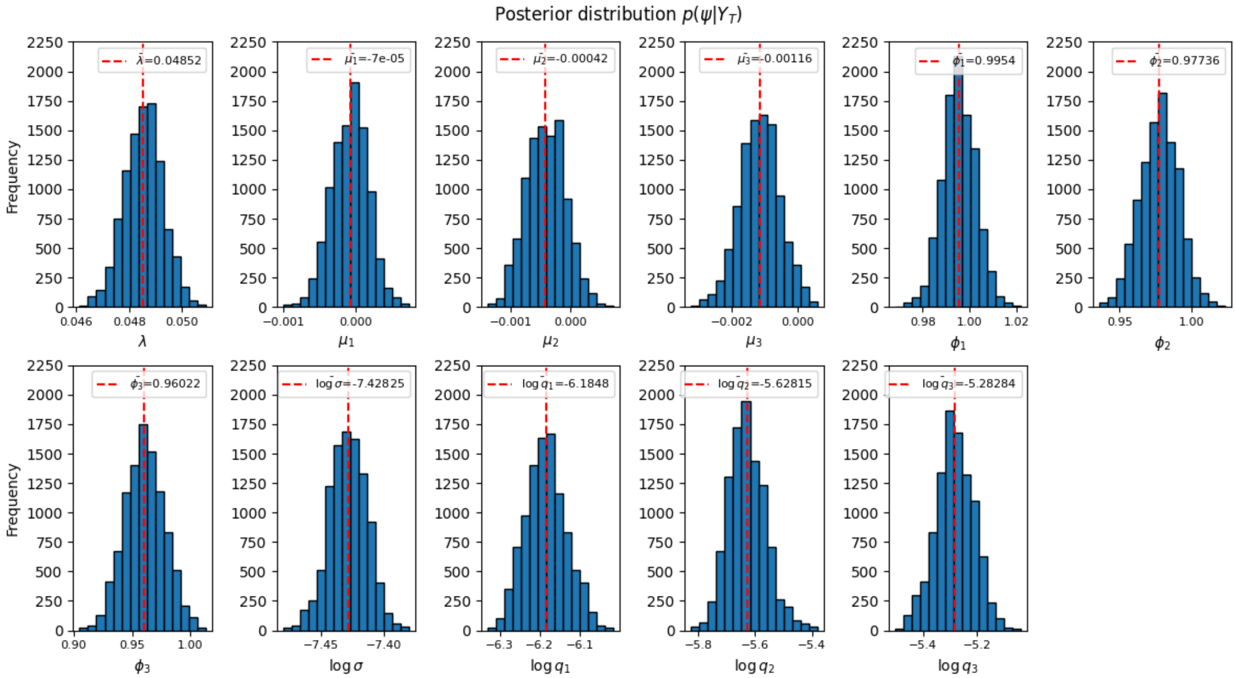


Figure 7.17: Histogram with the mean of the posterior distribution for each parameter of the DNS-SN model.

Table 7.18: The results of the Geweke diagnostic test for the chains of each parameter of the DNS-SN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2
Statistic $\|G\|$	0.10725	0.11116	0.25567	0.11244	0.10288	0.18105
Means differ significantly?	No	No	No	No	No	No
Parameter ψ_i	ϕ_3	σ	q_1	q_2	q_3	
Statistic $\|G\|$	0.16364	0.01710	0.26996	0.29633	0.47054	
Means differ significantly?	No	No	No	No	No	

Table 7.19: The MAPE values $\hat{\boldsymbol{\psi}}^{MAPE}$ for the DNS-SN model and the corresponding log-likelihood (LL), BIC and AIC values.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2
MAPE $\hat{\psi}_i^{MAPE}$	0.04862	-0.00006	-0.00044	-0.0011	0.99496	0.97722
Parameter ψ_i	ϕ_3	σ	q_1	q_2	q_3	
MAPE $\hat{\psi}_i^{MAPE}$	0.96224	0.00059	0.00203	0.0035	0.00501	
Model fit measure	LL	BIC	AIC			
Value	15392	-30723	-30762			

7.2.2 In-Sample Analysis

In this section we discuss the estimated states and provide some yield curve estimations of various dates as well as the residuals.

Estimated State Variables and Yields

We use the Kalman filter with the MLE values of the initial state $\hat{\boldsymbol{\beta}}^0$ and initial standard deviation p as in Table 7.14 and the MAPE values for the model parameters as in Table 7.19. The estimated state variables $\hat{\boldsymbol{\beta}}_t$ until November 2021 are shown in Figure 7.20. We can see that the estimated states are almost the same as the benchmark DNS model. Consequently, the difference in the residuals for both models are between $10^{-5} - 10^{-4}$ and thus negligible. So, we refer to the in-sample analysis of the benchmark DNS model in Subsection 7.1.2.

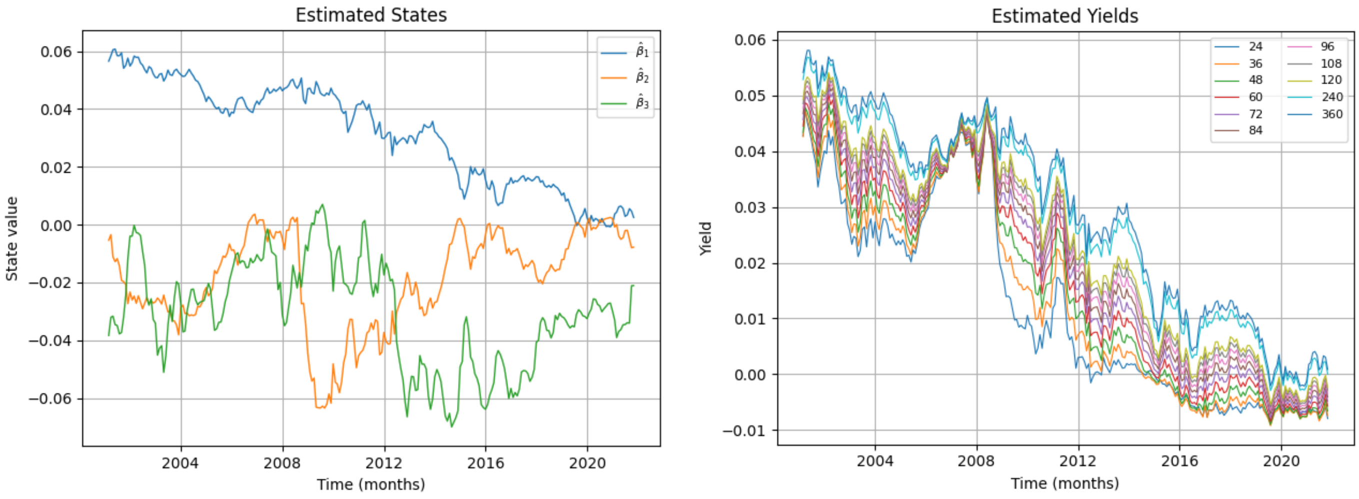


Figure 7.20: The estimated state variables $\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \hat{\beta}_{3,t}$ obtained by the Kalman filter and the estimated yields $\hat{\boldsymbol{y}}_t = \Lambda\hat{\boldsymbol{\beta}}_t$ for $t = 1, \ldots, 249$ with the DNS-SN model.

7.2.3 Forecasting Analysis

In this section we present the results of the one-month ahead forecast of the yield curve in December 2021 and in September 2008. For each maturity the posterior predictive values are simultaneously simulated 1000 times. Recall that the reason for modeling distinct state noise variances is because we expect that this might make the 95% credible region more accurate. In Figure 7.21 we can see that the 95% credible regions for both forecasts are indeed narrower than for the benchmark DNS model. The uncertainty captured by the observation noise $\pm\sigma$ is comparable with the benchmark DNS model, which is not surprising as the observation noise variance for the DNS-SN model is also relatively small compared with the state noise variances. Moreover, the DNS-SN model has difficulty in forecasting the shock in the short-term yields from August 2008 to September 2008 as well.

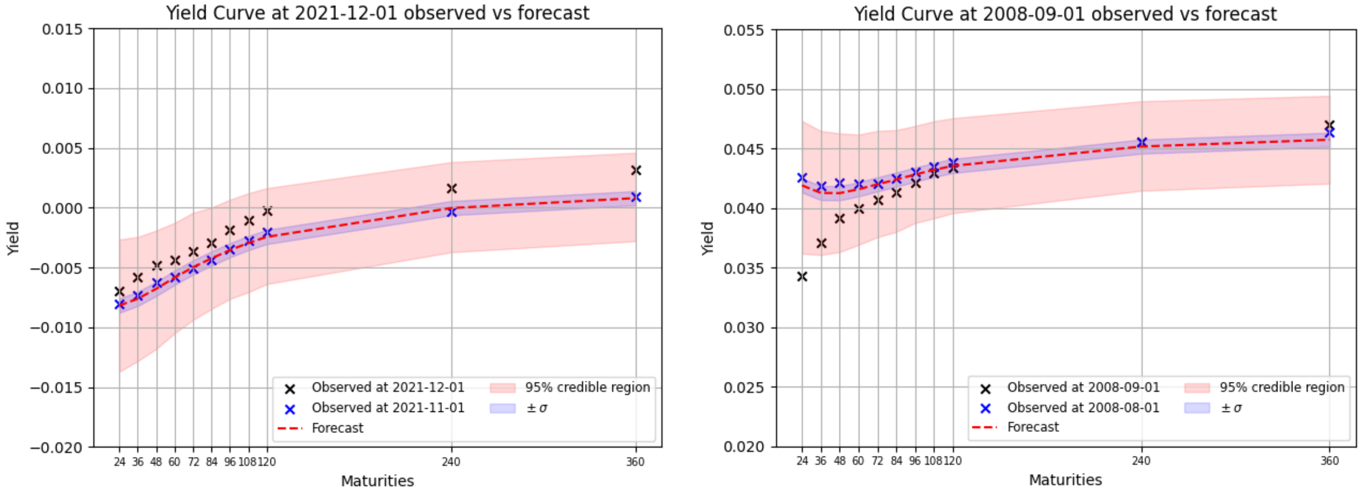


Figure 7.21: The one-month ahead forecast (dashed red line) of December 2021 (left) and September 2008 (right) with 95% credible regions (red surface) and the uncertainty due to the observation noise $\pm\sigma$ (blue surface).

7.3 Results of DNS-ARRW

In this section we present the parameter, in-sample and forecasting results for the DNS-ARRW model. Recall that this model has autoregressive (AR) observation noise and random walk (RW) state variables as specified in (6.27) and is given by

$$\begin{cases} \boldsymbol{y}_t = \begin{bmatrix}\Lambda & I\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_t\\ \boldsymbol{\varepsilon}'_t\end{bmatrix} + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0},\Sigma_\nu),\\ \begin{bmatrix}\boldsymbol{\beta}_t\\ \boldsymbol{\varepsilon}'_t\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu}\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}I & O\\ O^T & A\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_{t-1}\\ \boldsymbol{\varepsilon}'_{t-1}\end{bmatrix} + \begin{bmatrix}\boldsymbol{\eta}_t\\ \boldsymbol{\xi}_t\end{bmatrix}, & \begin{bmatrix}\boldsymbol{\eta}_t\\ \boldsymbol{\xi}_t\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\Sigma_\eta & O\\ O^T & \Sigma_\xi\end{bmatrix}\right), \end{cases} \tag{7.3}$$

with parameters $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, q, \sigma_\nu, \alpha_1, \dots, \alpha_{11}, \sigma_\xi\}$.

7.3.1 Parameter Estimation

The used scales and parameters resulting into convergence of the RWM algorithm are provided in Table 7.22 and 7.23. Notice that this model does not only have more parameters than the previous models, but it also has more state variables. The state variables are the three Nelson-Siegel variables $\boldsymbol{\beta}_t$ and the eleven autoregressive noise state variables $\boldsymbol{\varepsilon}'_t$. This means that the table with MLE values contains 14 additional parameter estimates compared with the parameters that are estimated with the RWM algorithm.

First we remark that the starting point 1×10^{-5} for σ_ν (ca. -11.51 for $\log \sigma_\nu$) seems to be so far off a high density region that it results into bad convergence for the other parameters as wel. The chain of $\log \sigma_\nu$ moves consistently towards around -8.8, so we have used a modified starting point of -8.8 (ca. 0.00015 for σ_ν). Then, we run the RWM algorithm with the MLE values as starting point for the rest of the parameters and the provided scales for 30000 iterations. The trace plot and histogram for each parameter is shown in Figure 7.24 and 7.25. We see interesting behaviour of the chains for the parameters. First, notice that the MLE as starting point for the DNS-ARRW model does not seem to be as good as in the previous models. For every parameter except for μ_1, μ_2, μ_3 and q the chains move significantly to other values. Especially the AR(1) parameters $\alpha_i,\ i = 1, \dots, 11$ and σ_ξ show a lot of movement from their original starting point. This shows one of the drawbacks of finding the MLE with a deterministic minimizer compared to exploring the parameter space stochastically with an MCMC method. It is likely that the used minimizer has encountered some local extremum. Consequently, we consider the first 10000 iterations as a so-called *burn-in period*, which are the iterations needed to reach the high density posterior region. Moreover, the RWM run has a average acceptance ratio of $\bar{\alpha} \approx 0.1636$, which indicates a reasonable rate of convergence close to 0.234. Moreover, we have performed the Geweke diagnostic test on the chains of each parameter with $\tau_A = 0.1, \tau_B = 0.5$ and a significance level of $\alpha = 0.05$, of which the results are provided in Table 7.26. The test indicates that the first and last segments of the chains for each parameter show no significant difference. So, overall we assume that the chains of each parameter has converged.

Table 7.22: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the DNS-ARRW model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	$\log q$	$\log \sigma_\nu$	α_1	α_2	α_3
Scale $\sigma_{i,RMW}$	0.00045	0.00013	0.00013	0.00013	0.0125	0.0328	0.018	0.0223	0.0175
Parameter ψ_i	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	$\log \sigma_\xi$
Scale $\sigma_{i,RMW}$	0.0155	0.0303	0.0293	0.0333	0.028	0.027	0.0093	0.0093	0.021

Table 7.23: The MLE values as RWM starting points $\boldsymbol{\psi}^{(0)}$ of the parameters approximated by the L-BFGS-B minimizer and the corresponding log-likelihood value for the DNS-ARRW model.

Parameter ψ_i	λ	$\hat{\beta}^0_1$	$\hat{\beta}^0_2$	$\hat{\beta}^0_3$	$\hat{\varepsilon}'^0_1$	$\hat{\varepsilon}'^0_2$	$\hat{\varepsilon}'^0_3$	$\hat{\varepsilon}'^0_4$	$\hat{\varepsilon}'^0_5$
Bounds	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$
Initial guess	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MLE value	0.04058	8×10^{-5}	-2×10^{-5}	-3×10^{-5}	-1×10^{-5}	-1×10^{-5}	-1×10^{-5}	-2×10^{-5}	-2×10^{-5}
Parameter ψ_i	$\hat{\varepsilon}'^0_6$	$\hat{\varepsilon}'^0_7$	$\hat{\varepsilon}'^0_8$	$\hat{\varepsilon}'^0_9$	$\hat{\varepsilon}'^0_{10}$	$\hat{\varepsilon}'^0_{11}$	p	μ_1	μ_2
Bounds	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$
Initial guess	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
MLE value	-2×10^{-5}	-2×10^{-5}	-2×10^{-5}	-2×10^{-5}	-1×10^{-5}	-2×10^{-5}	0.98332	-0.00023	2×10^{-5}
Parameter ψ_i	μ_3	q	σ_ν	α_1	α_2	α_3	α_4	α_5	α_6
Bounds	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$
Initial guess	0.0	0.1	0.1	0.5	0.5	0.5	0.5	0.5	0.5
MLE value	4×10^{-5}	0.00392	1×10^{-5}	0.57525	0.64171	0.61361	0.62992	0.54652	0.54793
Parameter ψ_i	α_7	α_8	α_9	α_{10}	α_{11}	σ_ξ			
Bounds	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$			
Initial guess	0.5	0.5	0.5	0.5	0.5	0.1			
MLE value	0.55042	0.62659	0.79915	0.64509	0.99683	0.00033			
Log-likelihood	16385								

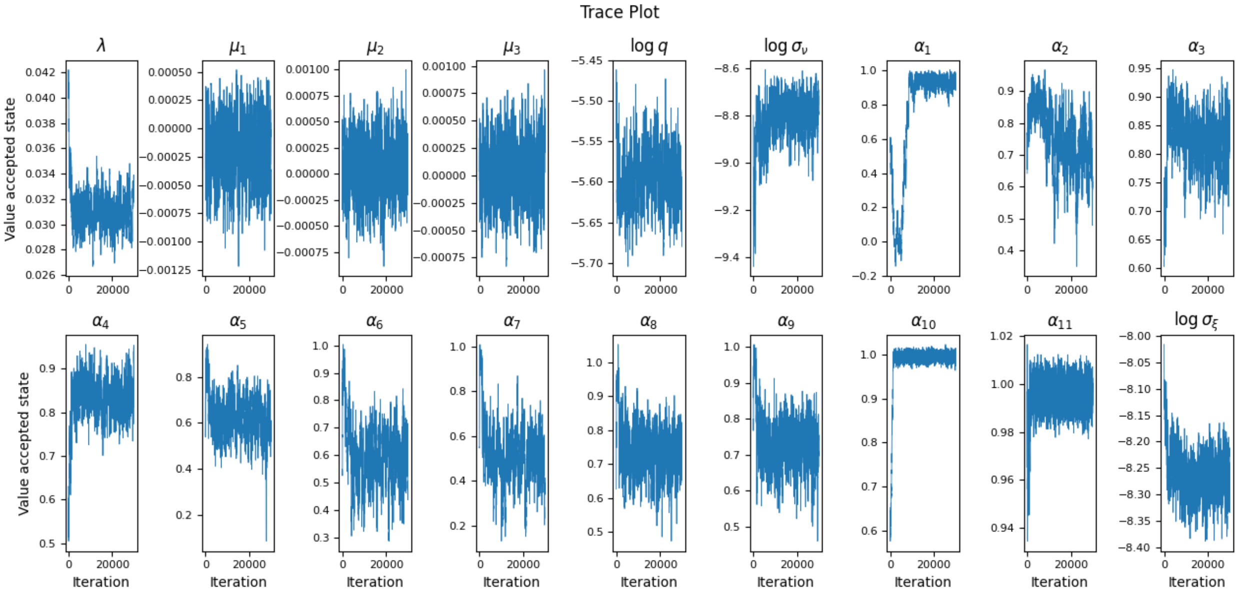


Figure 7.24: Trace plot of the chains for each parameter of the DNS-ARRW model.

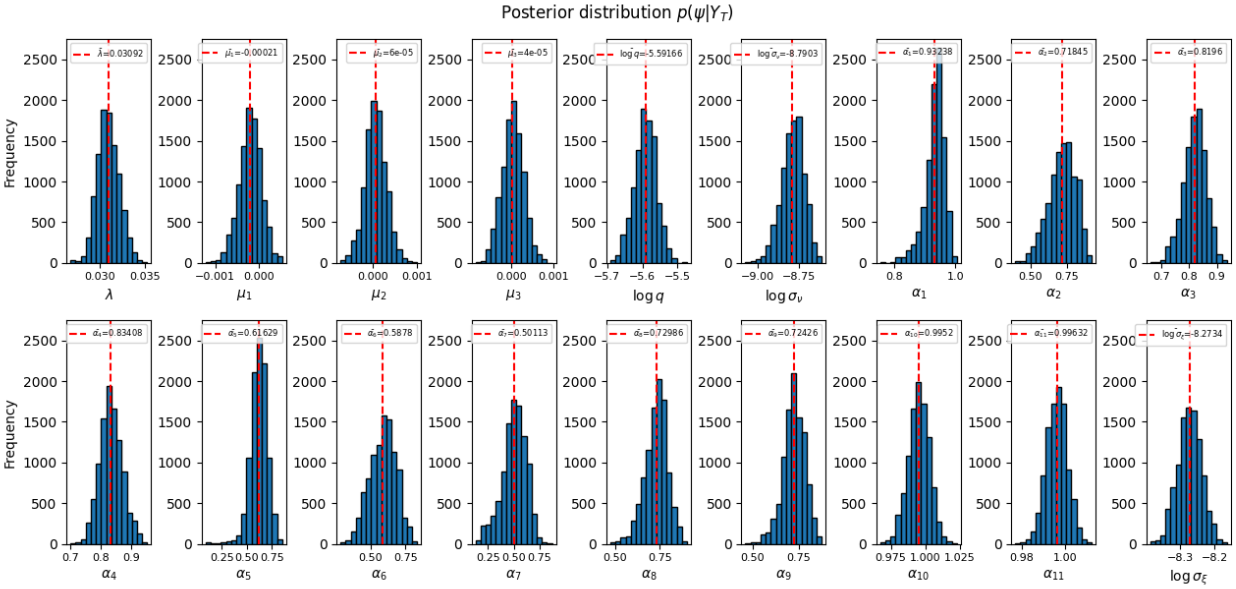


Figure 7.25: Histogram with the mean of the posterior distribution for each parameter of the DNS-ARRW model.

Table 7.26: The results of the Geweke diagnostic test for the chains of each parameter of the DNS-ARRW model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	q	σ_ν	α_1
Statistic $\lvert G\rvert$	0.20076	0.15447	0.13643	0.13241	0.11896	0.46536	0.56309
Means differ significantly?	No	No	No	No	No	No	No
Parameter ψ_i	α_2	α_3	α_4	α_5	α_6	α_7	α_8
Statistic $\lvert G\rvert$	0.76768	0.53971	0.26085	0.31863	0.10559	0.83157	0.00641
Means differ significantly?	No	No	No	No	No	No	No
Parameter ψ_i	α_9	α_{10}	α_{11}	σ_ξ			
Statistic $\lvert G\rvert$	0.15668	0.06212	0.07709	0.37153			
Means differ significantly?	No	No	No	No			

We use the MAPE again as the set of parameters for the in-sample analysis. Since the MLE does not seem to be as close to the highest density region of the posterior as for the previous models, we see that the MAPE is different from the MLE. The MAPE is provided in Table 7.27. Notice that we provide the exponential transformation of MAPE value for the log-transformed parameters. It seems that the shortest and two longest maturities show the most persistent "memory" of the past state values, since the three autoregression parameters α_1, α_{10} and α_{11} are above 0.9. For the longest maturities this seems quite intuitive, because longer-term maturities tend to change more gradually as those bond yields are associated with longer time horizons. However, it is quite surprising that the shortest-term maturity also shows such persistence of past values in the process. One would expect that the shortest-term maturities are more prone to news and unexpected short term developments. However, as we will see in the in-sample analysis the high values of α_1, α_{10} and α_{11} seem to be mainly the result of the original Nelson-Siegel state variables $\boldsymbol{\beta}_t$ having difficulty with modeling the short and long-term ends of the yield curve.

Table 7.27: The MAPE values $\hat{\boldsymbol{\psi}}^{MAPE}$ for the DNS-ARRW model and the corresponding log-likelihood (LL), BIC and AIC values.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	q	σ_ν	α_1
MAPE $\hat{\psi}_i^{MAPE}$	0.03163	-0.0002	-3.2×10^{-6}	-2.2×10^{-5}	0.00374	0.00015	0.94262
Parameter ψ_i	α_2	α_3	α_4	α_5	α_6	α_7	α_8
MAPE $\hat{\psi}_i^{MAPE}$	0.61878	0.79963	0.84313	0.58212	0.59090	0.53659	0.70086
Parameter ψ_i	α_9	α_{10}	α_{11}	σ_ξ			
MAPE $\hat{\psi}_i^{MAPE}$	0.71697	0.99686	0.99724	0.00026			
Model fit measure	LL	BIC	AIC				
Value	16457	-32815	-32878				

7.3.2 In-Sample Analysis

In this subsection we discuss the in-sample results based on the MAPE parameters in the previous subsection.

Estimated State Variables and Yields

Using the Kalman filter with the MAPE values for the model parameters and the MLE values as initial values for the Kalman filter we obtain the estimated state variables $(\hat{\boldsymbol{\beta}}_t, \hat{\boldsymbol{\varepsilon}}'_t)$. In Figure 7.28 the estimated state variables are shown and in Figure 7.29 the estimated yields resulting from the estimated states are shown.

It seems that the additional observation noise states $\hat{\boldsymbol{\varepsilon}}'_t$ mostly affect the state variable $\hat{\beta}_{2,t}$ associated with the slope of the yield curve. The slope state variable moves more gradually and does not reach zero after a short period around 2008. So, it looks like the observation noise states $\hat{\boldsymbol{\varepsilon}}'_t$ adopt some of the slope effects on the yield curve. Additionally, the observation noise states $\hat{\boldsymbol{\varepsilon}}'_t$ are most pronounced for the longest two maturities associated with $\hat{\varepsilon}'_{10,t}$ and $\hat{\varepsilon}'_{11,t}$ after 2008, which corresponds with the smaller $\hat{\beta}_{2,t}$ values after 2008. Between 2010 and 2014 the shortest-term maturity of 24 months associated with $\hat{\varepsilon}'_{1,t}$ is more pronounced. Moreover, it looks like the periods of higher volatility of 2008-2012 and 2020-2021 correspond mainly with the longest maturities $\hat{\varepsilon}'_{10,t}, \hat{\varepsilon}'_{11,t}$ and to a lesser extent $\hat{\varepsilon}'_{1,t}$. This seems to be mainly due to the fact that the original state variables $\beta_i,\ i = 1, 2, 3$ have difficulty in modeling the short and long ends of the yield curves. So, the observation noise state variables for those maturities $\hat{\varepsilon}'_{1,t}, \hat{\varepsilon}'_{10,t}$ and $\hat{\varepsilon}'_{11,t}$ need to be more pronounced and as a result have higher AR(1) parameters $\alpha_1, \alpha_{10}, \alpha_{11}$. This becomes more obvious if we look at the estimated yield curves in Figure 7.30. Notice that the term $\hat{\boldsymbol{\varepsilon}}'_t$ is not dependent of the maturity τ in the observation equation as opposed to the term $\Lambda\hat{\boldsymbol{\beta}}_t$, in which entries of Λ are dependent on τ. This means that we cannot easily vary τ to obtain the yield for maturities between the fixed maturities $\tau_1 = 24, \ldots, \tau_{11} = 360$ as with the benchmark DNS and DNS-SN model. So, the estimated yield curves in Figure 7.30 are based on the estimated $\hat{\boldsymbol{\beta}}_t$, while the actual yield estimates for the observed maturities are based on all state variables and provided as well for completeness. The yield curves that are shown are at the three dates where $\hat{\varepsilon}'_{1,t}, \hat{\varepsilon}'_{10,t}$ and $\hat{\varepsilon}'_{11,t}$ are the most extreme. In Figure 7.30 the observed yield curve is shown together with the estimated yield curve in July 2017 (minimum of $\hat{\varepsilon}'_{10,t}$), December 2013 (minimum of $\hat{\varepsilon}'_{11,t}$) and January 2011 (maximum of $\hat{\varepsilon}'_{1,t}$). We can see that the yield estimates with only the state variables $\hat{\boldsymbol{\beta}}$ are most accurate for maturities up to 120 months, but the estimates for the long-term maturities of 240 and 360 months are quite off without $\hat{\boldsymbol{\varepsilon}}'_t$.

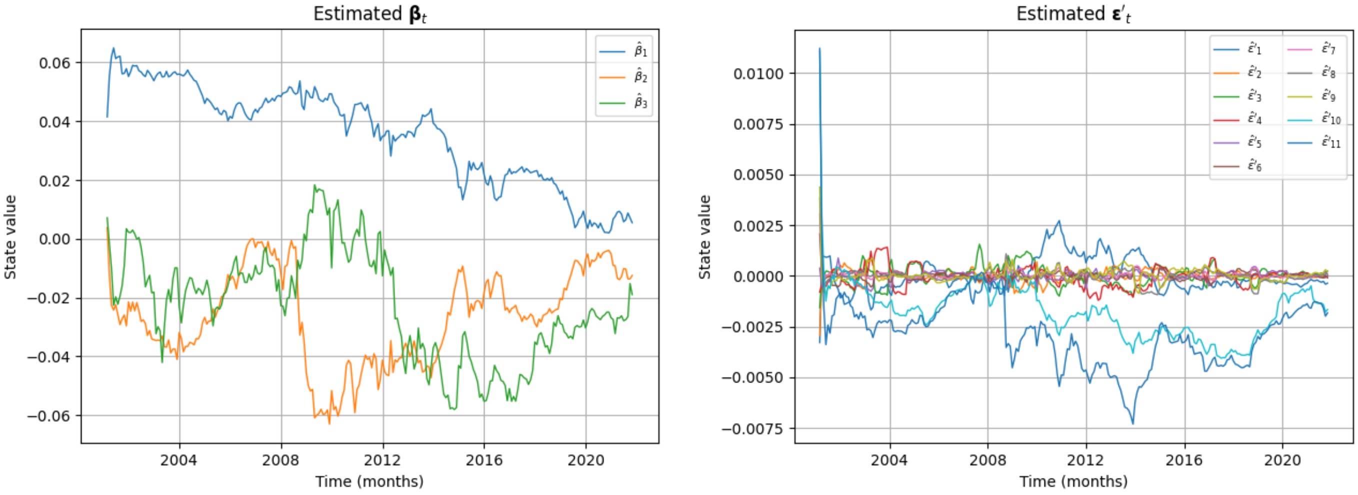


Figure 7.28: The estimated state variables $\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \hat{\beta}_{3,t}$ and $\hat{\varepsilon}'_{1,t}, \ldots, \hat{\varepsilon}'_{11,t}$ for $t = 1, \ldots, 249$ of the DNS-ARRW model obtained by the Kalman filter.

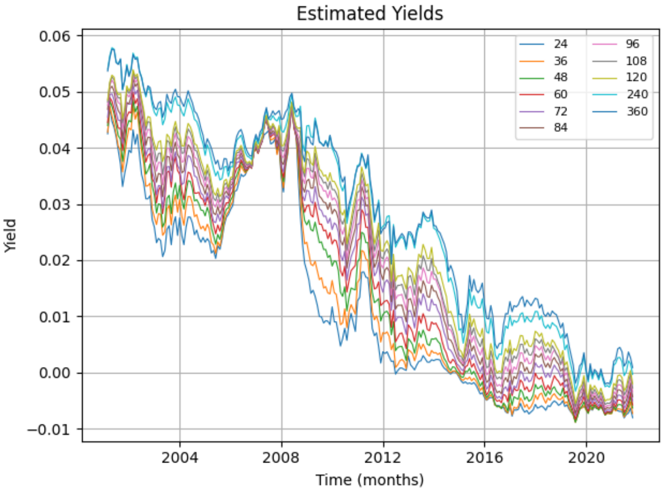


Figure 7.29: The estimated yields $\hat{\boldsymbol{y}}_t = \Lambda\hat{\boldsymbol{\beta}}_t + \hat{\boldsymbol{\varepsilon}}'_t$ for $t = 0, \ldots, 249$ for the DNS-ARRW model.

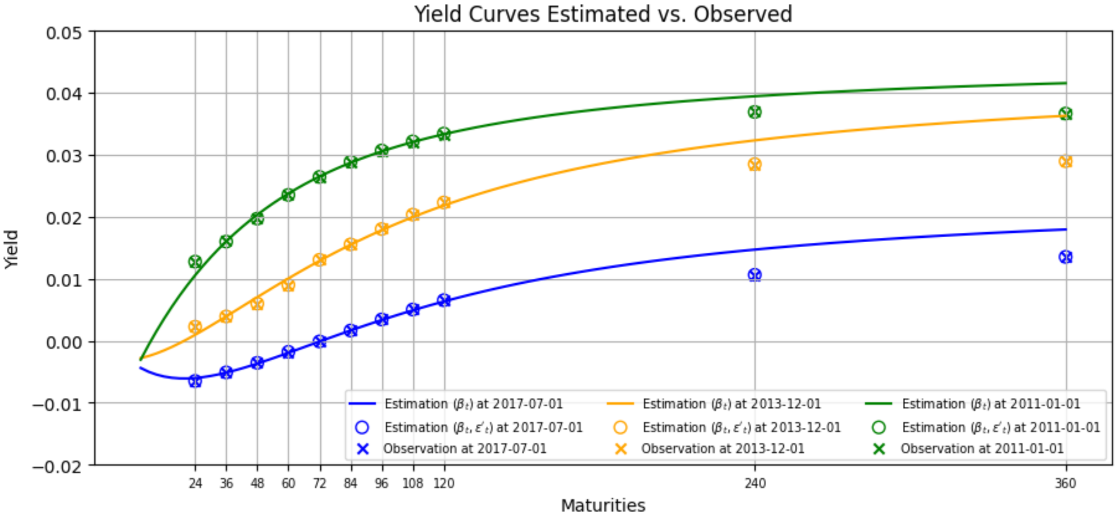


Figure 7.30: The in-sample observed (crosses) and estimated yield curves based on the state variables $\hat{\boldsymbol{\beta}}_t$ (smooth curves) and based on all states (circles) in July 2017 (blue), December 2013 (orange) and January 2011 (green). The estimated yields are based on the DNS-ARRW state variables inside the parentheses in the legend.

Residuals Analysis

The residuals of the DNS-ARRW model are shown for each maturity in Figure 7.31. Recall that we have modeled the observation noise $\boldsymbol{\nu}_t$ as a Gaussian white noise, denoted by $\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\nu)$. For the previous models this did not seem as a realistic assumption, because the residuals showed significant serial correlation. However, for the DNS-ARRW model we essentially model part of the observation noise as an AR(1) process ($\boldsymbol{\varepsilon}'_t$), so we would expect that the remainder, $\boldsymbol{\nu}_t$, resembles a white noise more. The plot of the residuals shows that the order of the error is at least one order of magnitude smaller than the residuals of the benchmark DNS and DNS-SN model.

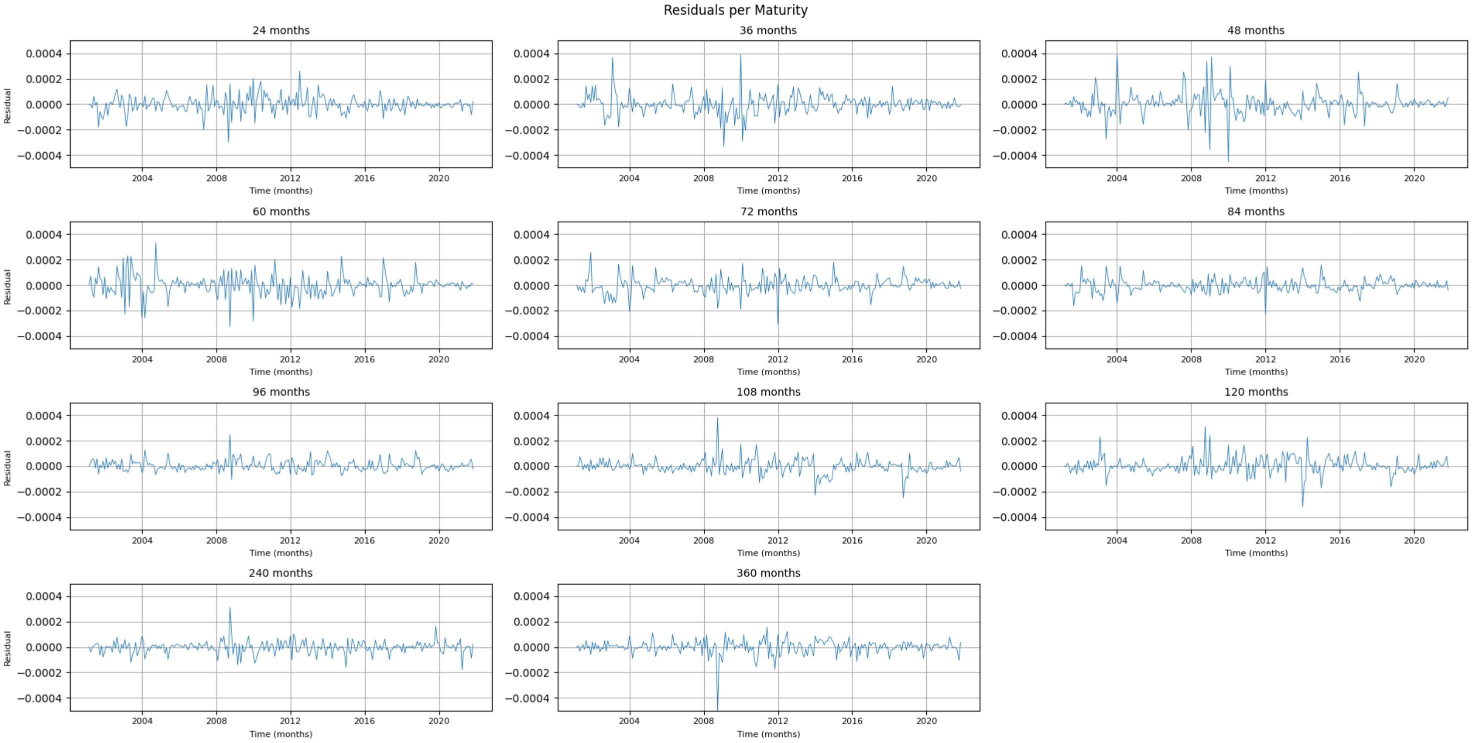


Figure 7.31: Residuals of the estimated yields for each maturity with the DNS-ARRW model.

In order to know whether the residuals are governed by a white noise process, we perform the Ljung-Box test again with a significance level of $\alpha = 0.05$ and lag $h = 1$. The results of the Ljung-Box test are provided in Table 7.32. We see that the residuals of the DNS-ARRW model are indeed a white noise for every maturity except for maturity $\tau_8 = 108$ months. In order to keep the results a bit organized we only note that the ACF and PACF of the residuals for every maturity except for $\tau_8 = 108$ show only a significant correlation for lag zero as we would expect for a white noise. Then, we focus only on the ACF and PACF of the residuals for maturity $\tau_8 = 108$, which are shown in Figure 7.33 and notice that the ACF and PACF show a sharp decrease of correlation already after lag one. Although, the correlations at lag $h = 1, 2, 3$ show slightly significant correlations we can conclude that the residuals overall seem to be a white noise process.

Table 7.32: Results of the Ljung-Box test for serial correlation in the residuals of the estimated yields for each maturity with the DNS-ARRW model.

Maturity τ_i	24	36	48	60	72	84
p value	0.62737	0.71795	0.76524	0.24038	0.06023	0.13120
White noise?	Yes	Yes	Yes	Yes	Yes	Yes
Maturity τ_i	96	108	120	240	360	
p value	0.05813	0.00818	0.22076	0.89934	0.29605	
White noise?	Yes	No	Yes	Yes	Yes	

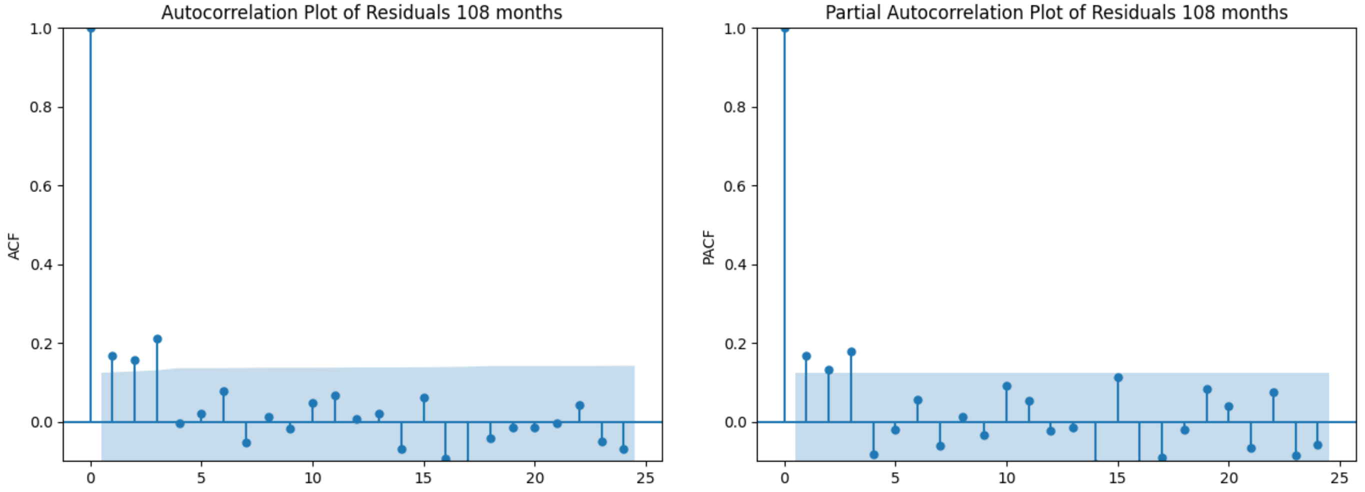


Figure 7.33: The ACF and PACF of the residuals of the estimated yields for the maturity of $\tau_8 = 108$ months.

7.3.3 Forecasting Analysis

In this subsection we dive into the forecasting analysis of the one-month ahead forecast of the yield curve in December 2021 and September 2008 for the DNS-ARRW model. For each maturity the posterior predictive value is simultaneously simulated 1000 times. The forecasts are presented in Figure 7.34. First, notice that the credible regions of the forecasts are quite wide. For the shortest-term maturity we see that 95% of the simulated yields are inside a bandwidth of ca. 150 bps, between -1.5% and 0.0%. So, although the DNS-ARRW model captures the yields that it has already seen well, it has more difficulty in accurately forecasting the “direction” of the yields compared with the benchmark DNS and DNS-SN models. Although, the forecast of the yield curve in September 2008 captures the shock in the the short-term maturity yields better due to the wider credible regions. However, it would be preferable if the accuracy of the short-term forecast would not improve at the cost of less accurate forecasts of the medium-term and long-term maturities. Subsequently, the standard deviation parameters σ_ν, σ_ξ and q have comparable orders of magnitude as their benchmark DNS counterparts σ and q, so it seems unlikely that so much additional variability in the simulations would stem from simulating (see steps 3 and 4 of Algorithm 3)

$$\boldsymbol{y}_{t+1} \sim \mathcal{N}(\Lambda \boldsymbol{\beta}_{t+1} + \boldsymbol{\varepsilon}'_{t+1}, \Sigma_\nu), \tag{7.4}$$

$$\begin{bmatrix} \boldsymbol{\beta}_{t+1} \\ \boldsymbol{\varepsilon}'_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} + \boldsymbol{\beta}_t \\ A\boldsymbol{\varepsilon}'_t \end{bmatrix}, \begin{bmatrix} \Sigma_\eta & O \\ O^T & \Sigma_\xi \end{bmatrix} \right). \tag{7.5}$$

Consequently, it seems more likely that the variability of the DNS-ARRW forecast originates from the relatively large uncertainty of the posterior samples of $\alpha_2, \ldots, \alpha_9$. It could be that the combination of a large sample variance in the posterior for mostly the autoregression parameters results into more variability in the simulated yields.

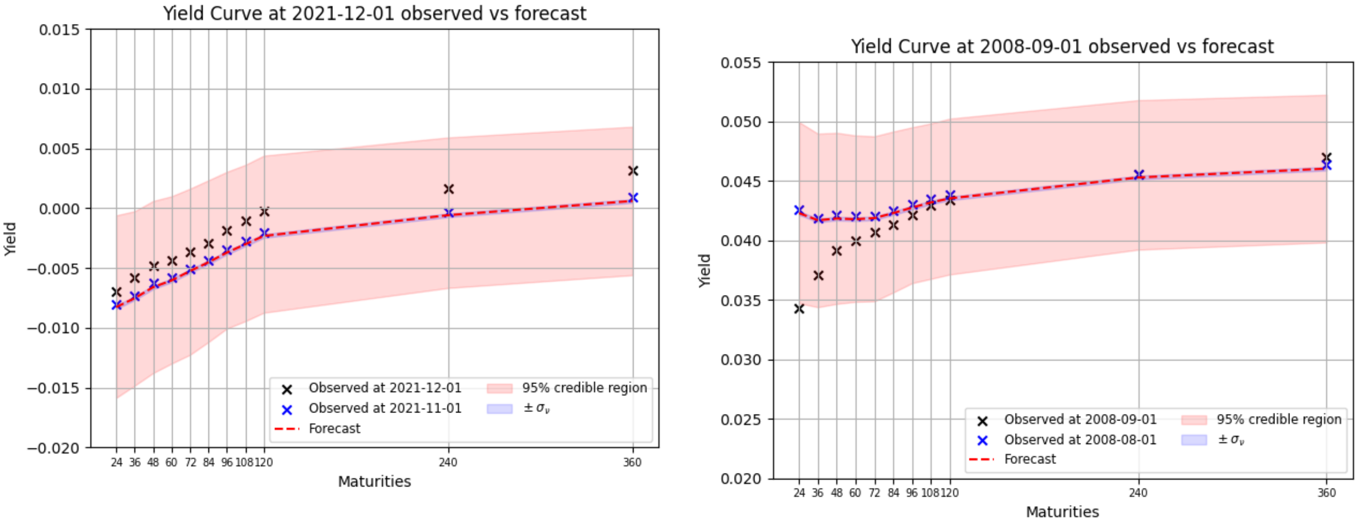


Figure 7.34: The one-month ahead forecast (dashed red line) of December 2021 (left) and September 2008 (right) with 95% credible regions (red surface) and the uncertainty due to the observation noise $\pm\sigma_\nu$ (blue surface).

Subsequently, we have already seen in the in-sample analysis in Figure 7.28 that there is quite some difference between the AR(1) processes of the short, medium and long-term maturities. Recall that we also assumed that all of these processes have equal variance by assuming they all have a white noise term distributed as $\mathcal{N}(0, \sigma_\xi^2)$. It seems that this assumption could be too restrictive as well and perhaps results into σ_ξ being some “average” value that fits the best for all maturities, but does not model the separate maturities well. Furthermore, notice that the uncertainty due to the observation noise σ_ν is even smaller for this model than the previous ones, which is the uncertainty of the observed yields one would expect from only the model. One way we could tackle this issue is comparable to how we modeled the distinct state noise terms for the DNS-SN model. We could introduce distinct state noise terms for this model as well. However, a drawback of this extension is that the resulting model would have 14 state noise variance parameters, which means 12 additional model parameters next to q and σ_ξ. In order to prevent adding perhaps too much parameters, we can assume that the various segments of the yield curve have equal variance. In particular, we can assume that the short-term (24-36 months), medium-term (36-108 months) and the long-term (120-360 months) segments of the yield curve have similar variances $\sigma_\xi^S, \sigma_\xi^M, \sigma_\xi^L$ respectively. As a result, we can define the noise variance Σ_ξ of the autoregressive $\varepsilon'_{1,t}, \ldots, \varepsilon'_{11,t}$ as

$$\Sigma_\xi = \begin{bmatrix} \Sigma_\xi^S & & \emptyset \\ & \Sigma_\xi^M & \\ \emptyset & & \Sigma_\xi^L \end{bmatrix} \in \mathbb{R}^{11\times 11}, \tag{7.6}$$

where $\Sigma_\xi^S = \text{diag}\left((\sigma_\xi^S)^2, (\sigma_\xi^S)^2\right) \in \mathbb{R}^{2\times 2}$, $\Sigma_\xi^M = \text{diag}\left((\sigma_\xi^M)^2, \ldots, (\sigma_\xi^M)^2\right) \in \mathbb{R}^{6\times 6}$ and $\Sigma_\xi^L = \text{diag}\left((\sigma_\xi^L)^2, \ldots, (\sigma_\xi^L)^2\right) \in \mathbb{R}^{3\times 3}$. We will use this approach of grouping the short, medium and long-term maturity segments of the yield curve for the last model DNS-OVOSN in Section 7.5 when modeling the observation noise covariance Σ_ε^+ and the volatility loadings Γ.

7.4 Results of DNS-OV

In this section we discuss the results of the parameter estimation and the in-sample and forecasting analysis for the DNS model with volatility modeled through the observation noise, DNS-OV in short, as specified in (6.28). Recall that this model is given by

$$\begin{cases} \boldsymbol{y}_t = \begin{bmatrix} \Lambda & \Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} + \boldsymbol{\varepsilon}_t^+, & \boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+), \\ \begin{bmatrix} \boldsymbol{\beta}_t \\ \varepsilon_t^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{t-1} \\ \varepsilon_{t-1}^* \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix}, & \begin{bmatrix} \boldsymbol{\eta}_t \\ \varepsilon_t^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_\eta & 0 \\ 0 & h_t \end{bmatrix}\right), \\ h_t = \gamma_0 + \gamma_1 (\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1}, \end{cases} \tag{7.7}$$

with parameters $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma^+, q, \gamma_0, \gamma_1, \gamma_2, \Gamma_1, \ldots, \Gamma_{11}\}$.

7.4.1 Parameter Estimation

In this subsection we provide the scales and starting points that result into convergence of the RWM algorithm. Like the DNS-ARRW model, the DNS-OV model has an additional state variable. This means that we have to find starting points for four state variables, the original state variables of the Nelson-Siegel model $\boldsymbol{\beta}_t$ and the "common shock" state variable ε_t^*. Recall that the term $(\varepsilon_{t-1}^*)^2$ is approximated by $((\hat{\varepsilon}^*)_{t-1}^{t-1})^2 + (p_\varepsilon)_{t-1}^{t-1}$. So, the approximation of the GARCH process is dependent on the variance estimation $(p_\varepsilon)_{t-1}^{t-1}$ of the common shock state variable ε_t^*. Consequently, due to the possible sensitivity of the GARCH process to the value of $(p_\varepsilon)_{t-1}^{t-1}$ we define the initial covariance matrix as $P_0^0 = \text{diag}(p_1^0, p_2^0, p_3^0, p_\varepsilon^0)$ to have a more accurate initial state when employing the Kalman filter. Furthermore, recall that we have fixed $\gamma_0 = 0.0001$, so this parameter is not taken into account when we estimate parameters. In this subsection we first discuss the RWM run with the volatility loadings $\Gamma_1, \ldots, \Gamma_{11}$ with scales as in Table 7.36 and the MLE values as starting points shown in Table 7.35 except for γ_1 and γ_2, for 30000 iterations. Particularly, the starting points for γ_1 and γ_2 are set to 0.7 and 0.05 respectively for better convergence as the chains tend to those values starting from their MLE value. Next, we will discuss the RWM run without the volatility loadings with scales as in Table 7.38 and the same starting points as the run with the volatility loadings, which has been run for 50000 iterations.

Table 7.35: The MLE values as RWM starting points $\boldsymbol{\psi}^{(0)}$ of the parameters approximated by the L-BFGS-B minimizer and the corresponding log-likelihood value for the DNS-OV model.

Parameter ψ_i	λ	$\hat{\beta}_1^0$	$\hat{\beta}_2^0$	$\hat{\beta}_3^0$	$\hat{\varepsilon}^{*0}$	p_1^0	p_2^0	p_3^0	p_ε^0
Bounds	$(10^{-4},\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$
Initial guess	0.1	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0
MLE value	0.04037	1×10^{-5}	0.0	-1×10^{-5}	0.0	0.9999	0.99986	0.9998	1.0
Parameter ψ_i	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	σ^+	q	γ_1
Bounds	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(-\infty,\infty)$	$(10^{-4},\infty)$	$(10^{-4},\infty)$	$(10^{-4},1)$
Initial guess	0.0	0.0	0.0	0.99	0.99	0.99	0.01	0.01	0.45
MLE value	-3×10^{-5}	-0.00044	-0.00069	0.99333	0.97646	0.97039	0.00043	0.0028	0.43733
Parameter ψ_i	γ_2	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Γ_6	Γ_7	Γ_8
Bounds	$(10^{-4},1)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$
Initial guess	0.45	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
MLE value	0.43075	0.04923	0.05653	0.07105	0.08012	0.09581	0.10413	0.11165	0.11645
Parameter ψ_i	Γ_9	Γ_{10}	Γ_{11}						
Bounds	$(10^{-5},\infty)$	$(10^{-5},\infty)$	$(10^{-5},\infty)$						
Initial guess	0.1	0.1	0.1						
MLE value	0.12224	0.11382	0.08914						
Log-likelihood	15850								

Random Walk Metropolis with Volatility Loadings

Recall that we already gave a preview of the slow convergence of the volatility loadings $\Gamma_1, \dots, \Gamma_{11}$ in Section 6.3.1. In Figure 7.37 the matrix correlation plot of the volatility loadings $\Gamma_1, \dots, \Gamma_{11}$ are shown for a RWM run with scales as in Table 7.37 and starting points as mentioned at the beginning of this section. Then, we can see the high mutual correlation between $\Gamma_1, \dots, \Gamma_{11}$, which is increasing as the maturities get closer to each other. Consequently, we fix these parameters at the maximum likelihood estimation values as provided in Table 7.35 and we focus mainly on the GARCH(1,1) parameters γ_1 and γ_2 in order to model parameter uncertainty of the volatility process. So, the results of the in-sample and forecasting analysis are based on the posterior approximation of the remainder of the parameters and $\Gamma_1, \dots, \Gamma_{11}$ fixed as we will see in the remaining part of this section. Finally, we note that the average acceptance ratio for this RWM run is $\bar{\alpha} \approx 0.1047$, which is already quite low with the chains of the volatility loadings that still have to converge.

Table 7.36: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the DNS-OV model for a RWM run with the volatility loadings $\Gamma_1, \dots, \Gamma_{11}$.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma^+$
Scale $\sigma_{i,RMW}$	0.00023	0.00017	0.00017	0.00016	0.0038	0.0039	0.0038	0.0039
Parameter ψ_i	$\log q$	γ_1	γ_2	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5
Scale $\sigma_{i,RMW}$	0.0078	0.019	0.0065	0.00092	0.0009	0.00089	0.00089	0.00089
Parameter ψ_i	Γ_6	Γ_7	Γ_8	Γ_9	Γ_{10}	Γ_{11}		
Scale $\sigma_{i,RMW}$	0.00089	0.00089	0.00091	0.00091	0.00093	0.00095		

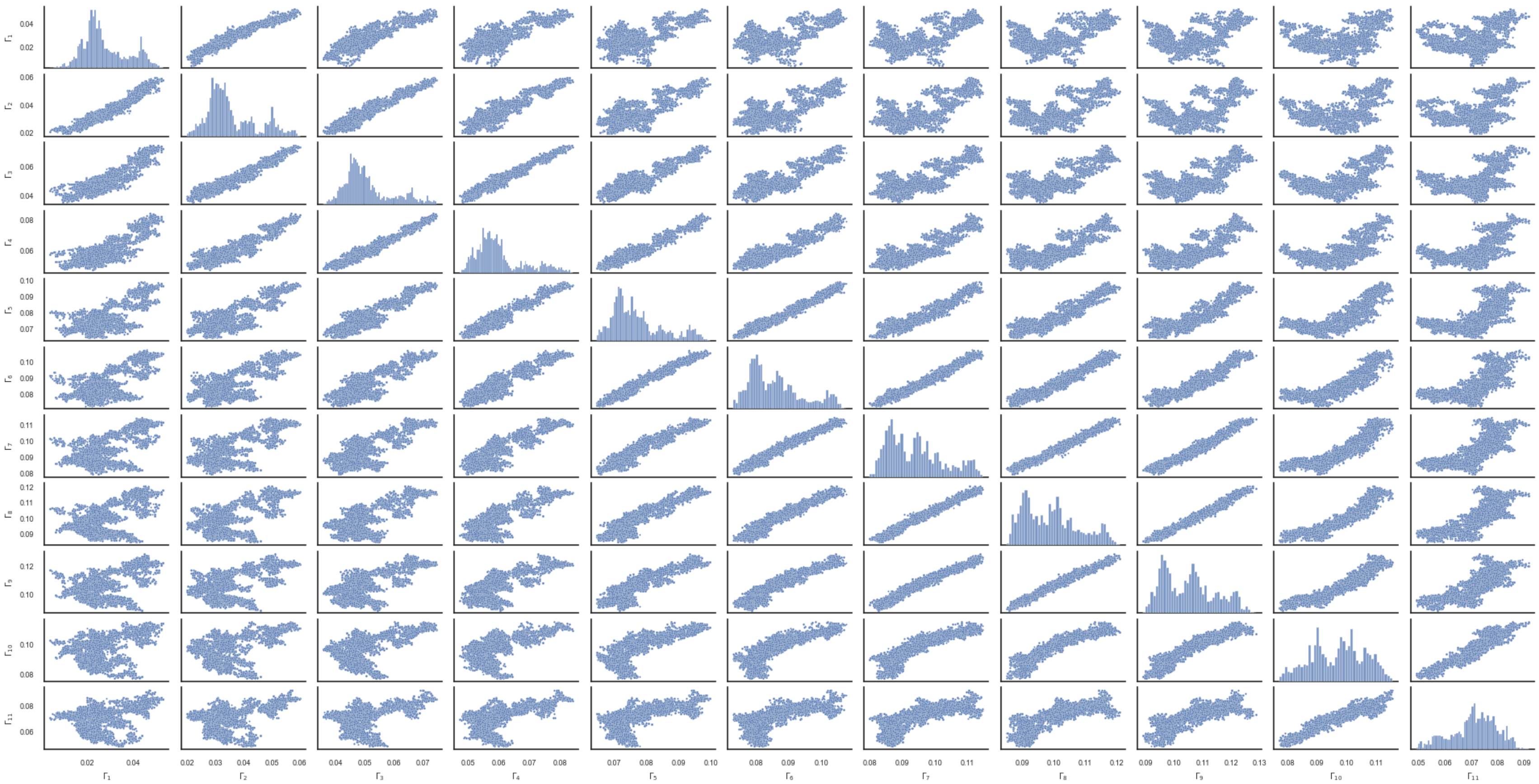


Figure 7.37: The matrix correlation plot of the samples of the DNS-OV volatility loading parameters $\Gamma_1, \ldots, \Gamma_{11}$ from a RWM run with 30000 iterations.

Random Walk Metropolis without Volatility Loadings

Recall that the starting points for the RWM run without the volatility loadings are the same as with the volatility loadings. The used scales resulting into convergence are provided in Table 7.38.

Table 7.38: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the DNS-OV model without the volatility loadings $\Gamma_1, \ldots, \Gamma_{11}$.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma^+$
Scale $\sigma_{i,RMW}$	0.00025	0.00018	0.0002	0.00018	0.0045	0.006	0.005	0.009
Parameter ψ_i	$\log q$	γ_1	γ_2					
Scale $\sigma_{i,RMW}$	0.013	0.025	0.012					

First, we note that the average acceptance ratio of this RWM run is $\bar{\alpha} \approx 0.2539$, which indicates a good rate of convergence. The trace plots of each parameter is shown in Figure 7.39 and shows reasonable white noise patterns for every parameter. In addition, the corresponding histograms for each parameter are shown in Figure 7.40. Particularly, the results for the parameters $\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma^+, q$ are similar to the previous models as they concentrate narrowly around similar mean values. On the contrary, the results for the GARCH$(1,1)$ parameters γ_1, γ_2 show a relatively high sample variance as 95% of the samples for the parameters are in the range $\gamma_1 \in (0.615, 0.899)$ and $\gamma_2 \in (0.003, 0.132)$. Recall that γ_1, γ_2 indicate the amount of volatility that can be attributed to either the common shock process ε_t^* or to the volatility h_t itself respectively as the GARCH$(1,1)$ process is given by

$$h_t = 0.0001 + \gamma_1 \left(\varepsilon_{t-1}^*\right)^2 + \gamma_2 h_{t-1}. \tag{7.8}$$

Then, the uncertainty for these parameters could be due to the yield observations not providing strong enough evidence for a particular value of γ_1 and to a lesser extent γ_2. Perhaps the monthly yield data exhibits too little volatility to provide precise estimates for these parameters, leading to a wider range of plausible values. Additionally, the results of γ_1 and γ_2 are also in contrast with the estimated values of Koopman et al. (2010), which are $\gamma_1 \approx 0.471$ and $\gamma_2 \approx 0.506$. In our case the volatility seems to originate mainly from the common shock process ε_t^* and to a small extent from the volatility process h_t itself. This means that the estimated volatility is prone to changes in the common shock process, but the effects do not persist for a longer period.

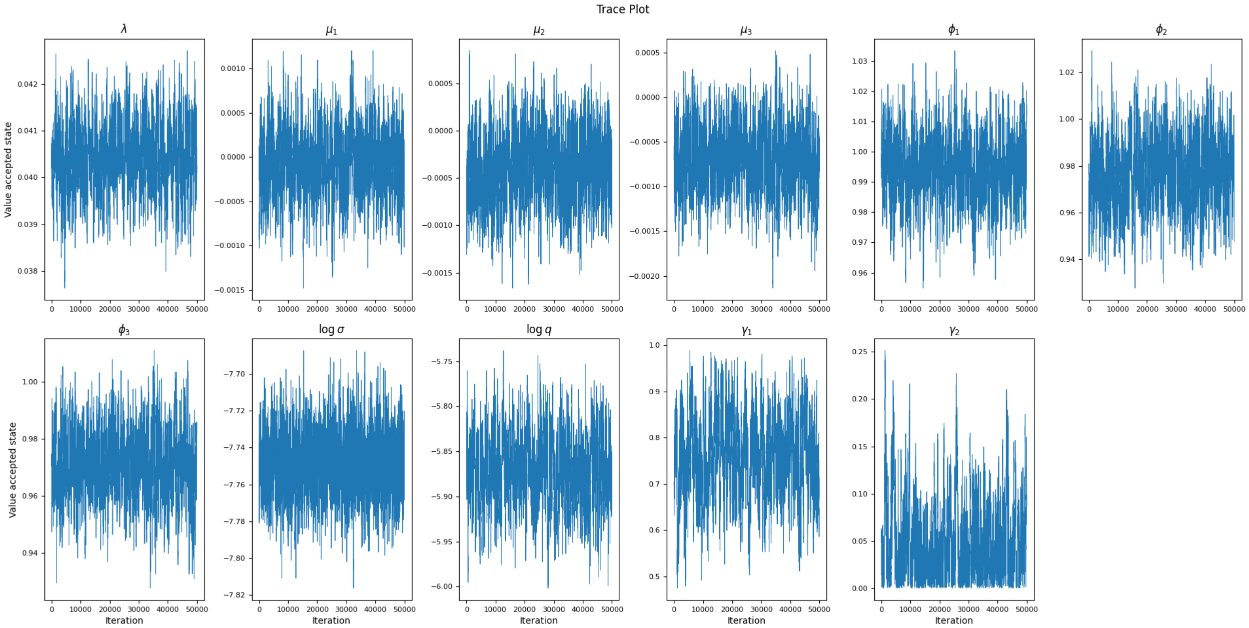


Figure 7.39: Trace plot of the chains for each parameter of the DNS-OV model.

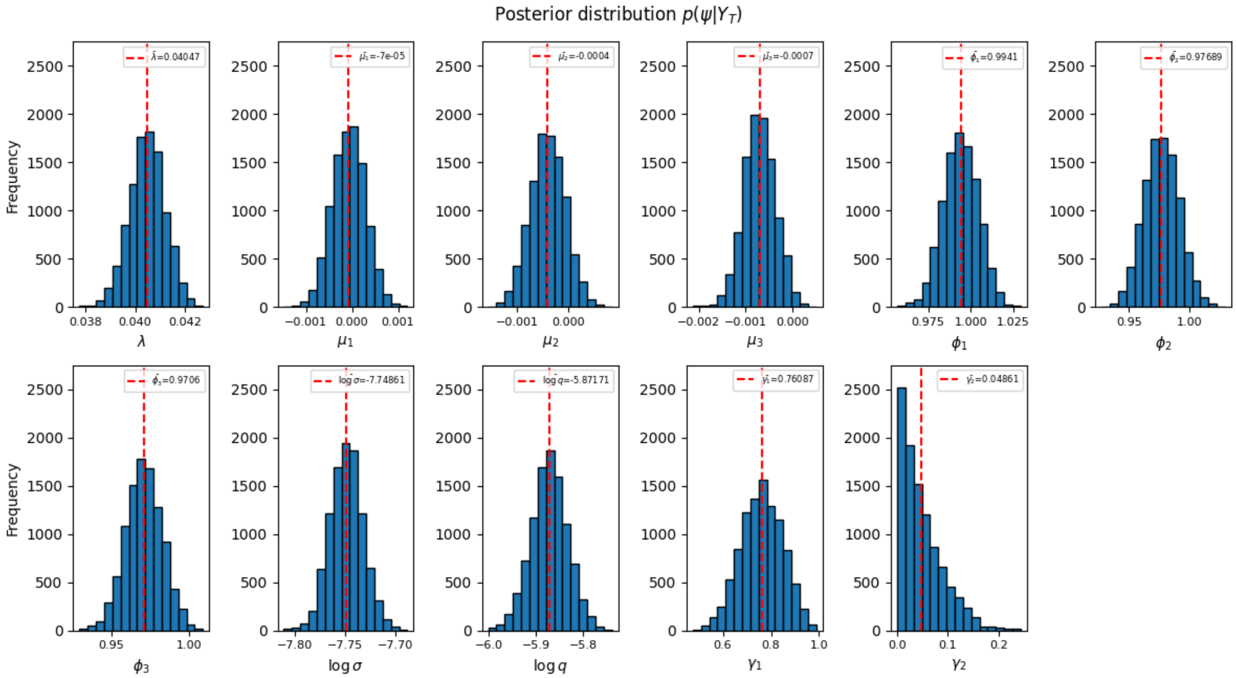


Figure 7.40: Histogram with the mean of the posterior distribution for each parameter of the DNS-OV model.

Moreover, we have performed the Geweke test on the chains of each parameter with $\tau_A = 0.1, \tau_B = 0.5$ and a significance level of $\alpha = 0.05$. The results of the test are provided in Table 7.41 and indicate that the means of the first and last segments of the chains are not significantly different. So, together with the results of the trace plot we assume that the chains of every parameter has converged. Furthermore, we will use the MAPE again for the in-sample analysis as provided in Table 7.42. Notice that the MAPE values result into a relatively high log-likelihood and better BIC and AIC values than the benchmark DNS, which are 15333, -30616 and -30648 respectively.

Table 7.41: The results of the Geweke diagnostic test for the chains of each parameter of the DNS-OV model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3
Statistic $\|G\|$	0.25450	0.01977	0.11419	0.03951	0.05494	0.11137	0.05938
Means differ significantly?	No	No	No	No	No	No	No
Parameter ψ_i	σ^+	q	γ_1	γ_2			
Statistic $\|G\|$	0.00094	0.05955	0.26693	0.29422			
Means differ significantly?	No	No	No	No			

Table 7.42: The MAPE values $\hat{\boldsymbol{\psi}}^{MAPE}$ for the DNS-OV model and the corresponding log-likelihood (LL), BIC and AIC values.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3
MAPE $\hat{\psi}_i^{MAPE}$	0.04068	1×10^{-6}	-0.00038	-0.0006	0.98866	0.97613	0.97465
Parameter ψ_i	σ^+	q	γ_1	γ_2			
MAPE $\hat{\psi}_i^{MAPE}$	0.00043	0.0028	0.81126	0.0036			
Model fit measure	LL	BIC	AIC				
Value	15850	-31573	-31654				

7.4.2 In-Sample Analysis

In this subsection we present the results for the in-sample analysis for the DNS-OV model. Notice that this model has an additional volatility related state variable, so we also provide an estimation of the volatility as opposed to the previous models.

Estimated State Variables and Yields

We use the Kalman filter with the additional estimation of the GARCH$(1,1)$ process as described in Subsection 5.2.2 to obtain the estimates of the state variables $(\hat{\boldsymbol{\beta}}, \hat{\varepsilon}^*)$. We have used the MLE values of the initial state $(\hat{\boldsymbol{\beta}}^0, (\hat{\varepsilon}^{*0}))$ and the initial standard deviations $p_1^0, p_2^0, p_3^0, p_\varepsilon^0$ for the Kalman filter as in Table 7.35 and the MAPE values as in Table 7.42.

The results of estimating the state variables and the yields are shown in Figure 7.43. It is interesting to see that the common shock state variable $\hat{\varepsilon}_t^*$ seems to affect the curvature $\hat{\beta}_{3,t}$ the most, compared to the estimated state variables of the benchmark DNS model. Moreover, we can see that the common shock process is most significant between 2008 and 2014. This period coincides with yield curves that decrease quite significantly and the range between the short and long-term maturity yields increases. In order to better grasp the behaviour of the common shock process $\hat{\varepsilon}_t^*$ we compare the yield curves in January 2009 and December 2010 of the first two peaks of $\hat{\varepsilon}_t^*$ with the yield curves in January 2003 and January 2020 with relatively small $\hat{\varepsilon}_t^*$ values, which are shown in Figure 7.44. We note that the yield curves are based on the state variables $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$ again, since it is difficult to interpolate the intermediate yields correctly with the additional $\Gamma\hat{\varepsilon}_t^*$ term similar to the DNS-ARRW model. Nevertheless, the yield curves still show which maturities and in which way the common shock process $\hat{\varepsilon}_t^*$ affects the most. For completeness, we have also provided the actual yield estimates based on all state variables. Then, we can see that the process $\hat{\varepsilon}_t^*$ is not very pronounced for quite flat yield curves as in January 2020. Moreover, it seems that $\hat{\varepsilon}_t^*$ is most pronounced if the yield curve has a large range of yields. Specifically, it seems that a large $\hat{\varepsilon}_t^*$ is the case when the yield curve has a quite steep medium-term segment as in January 2009 and December 2010. Although, the medium-term segment of the yield curve in January 2003 also shows some general steepness. In particular, the medium-term yields range ca. 80 bps between the medium-term maturities (48 vs 108 months) in January 2003, while that difference is around 120 and 150 bps in January 2009 and December 2010 respectively. This is in line with the common shock process $\hat{\varepsilon}_t^*$ mainly affecting the curvature $\hat{\beta}_{3,t}$, which has most effect on the medium-term maturities of the yield curve. So, it seems that $\hat{\varepsilon}_t^*$ models some underlying process affecting the volatility mainly through medium-term maturities.

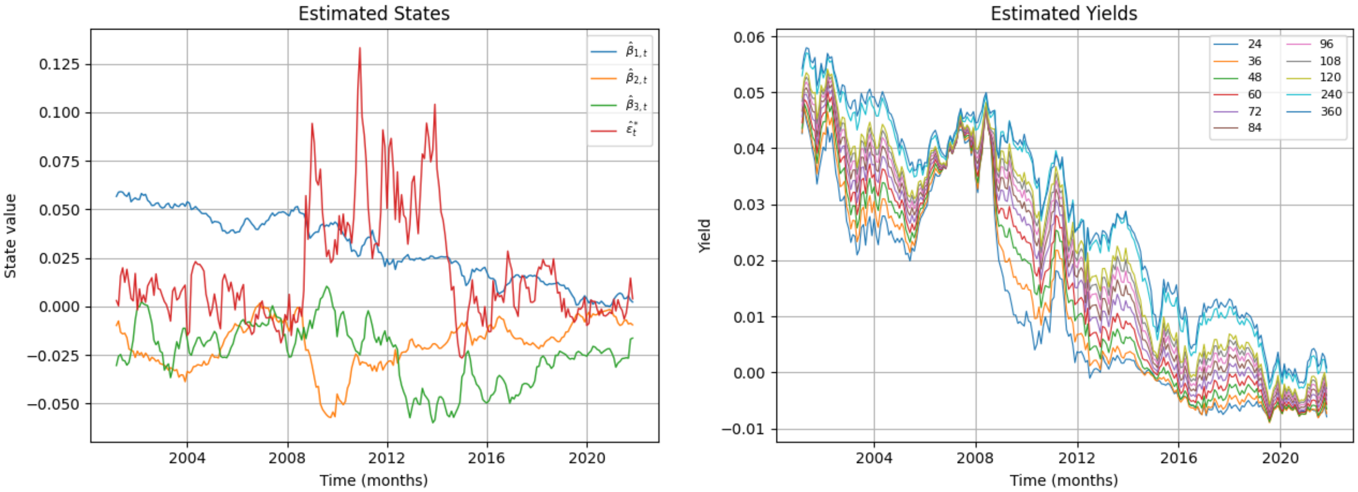


Figure 7.43: The estimated state variables $\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \hat{\beta}_{3,t}$ and $\hat{\varepsilon}_t^*$ for $t = 1, \ldots, 249$ obtained by the Kalman filter and the estimated yields $\hat{\boldsymbol{y}}_t = \Lambda \hat{\boldsymbol{\beta}}_t + \Gamma \hat{\varepsilon}_t^*$ with the DNS-OV model.

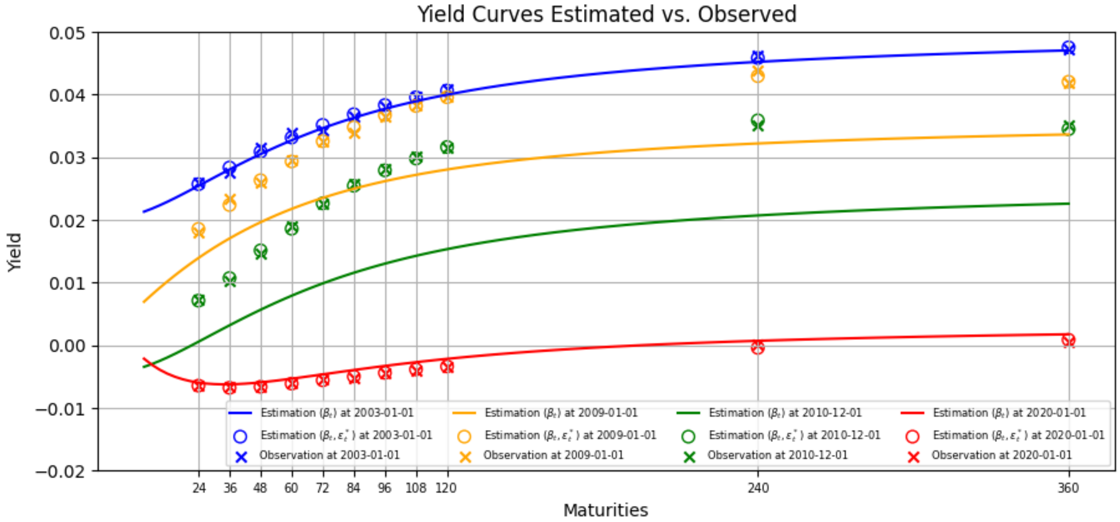


Figure 7.44: The in-sample observed (crosses) and estimated yield curves based on only the state variables $\hat{\boldsymbol{\beta}}_t$ (smooth curves) and based on all states (circles) in January 2003 (blue), January 2009 (orange), December 2010 (green) and January 2020 (red). The estimated yields are based on the DNS-OV state variables inside the parentheses in the legend.

Recall that the volatility for the yields of some maturity $y(\tau_i)$ is given by multiplying the squared volatility loading Γ_i^2, which is the maturity-specific volatility factor, with the volatility h_t, which is the variance of the common shock process $\hat{\varepsilon}_t^*$. In Figure the values of $\Gamma_1, \ldots, \Gamma_{11}$ of maturity $24, \ldots, 360$ respectively are presented. We can see that the volatility loading increases from a maturity of 24 months until it reaches the highest volatility loading at a maturity of 120 months before decreasing for the maturities of 240 and 360 months. This suggests that the common volatility h_t is more prominent for the medium-term maturities around 120 months and the long-term maturities. The volatility loadings are based on the entire in-sample interval from March 2001 to November 2021, so the values could indicate that over the whole period bond yields with a maturity around 120 months experience relatively more volatility than the shorter-term maturities. Then, in Figure 7.46 the volatility is shown for the yields with maturities 24, 72 and 360 months, given by $\Gamma_1^2 h_t, \Gamma_5^2 h_t$ and $\Gamma_{11}^2 h_t$ respectively. Notice that the volatility resembles the common shock process, which is not too surprising as γ_2 is close to zero. Consequently, this leaves the volatility h_t mainly affected by the term $\gamma_1 \left(\hat{\varepsilon}_{t-1}^*\right)^2$, which can be seen in the fact that h_t shows quite sudden peaks that do not persist longer through time.

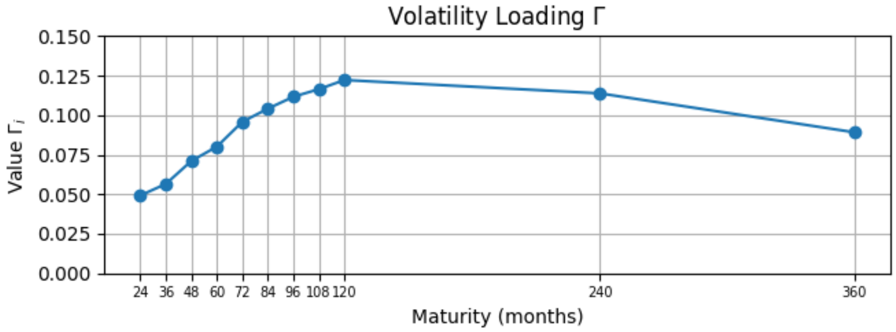


Figure 7.45: The values of the DNS-OV volatility loadings $\Gamma_1, \ldots, \Gamma_{11}$ for maturities $24, \ldots, 360$ months respectively.

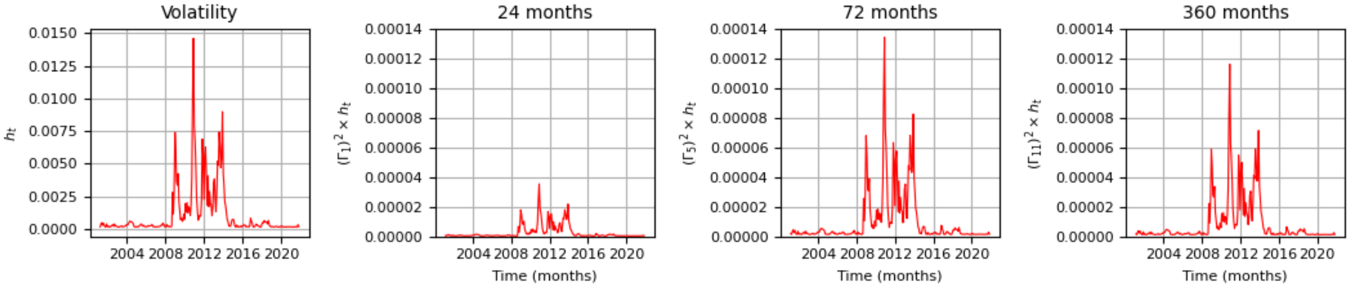


Figure 7.46: The volatility process h_t of the DNS-OV model and the volatilities $\Gamma_i^2 h_t$ for $i = 1, 5, 11$ corresponding to maturities of 24, 72 and 360 months respectively.

Residuals Analysis

The residuals of the yields of the DNS-OV model compared with the benchmark DNS model are shown for each maturity in Figure 7.47. Recall that the observation noise $\boldsymbol{\varepsilon}_t^+$ in this model is assumed to be a white noise distributed as $\boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+)$, so one would expect the residuals to show no serial correlation. We can see that the DNS-OV model estimates the yields more accurately for the for the longer-term maturities of 96 months to 360 months. Moreover, the residuals for the short-term and medium-term maturities of 24 to 84 months are estimated quite similarly to the benchmark DNS model. So, the greatest advantage of the DNS-OV model seems to be in the accuracy of the long-term maturities of the yield curve. Subsequently, we perform a Ljung-Box test with significance level $\alpha = 0.05$ and lag $h = 1$ to test whether the residuals show serial correlation. The results of the test are provided in Table 7.48. Notice that the p values for the residuals of each maturity is very low indicating that the residuals for the DNS-OV yield estimates are not a white noise process significantly.

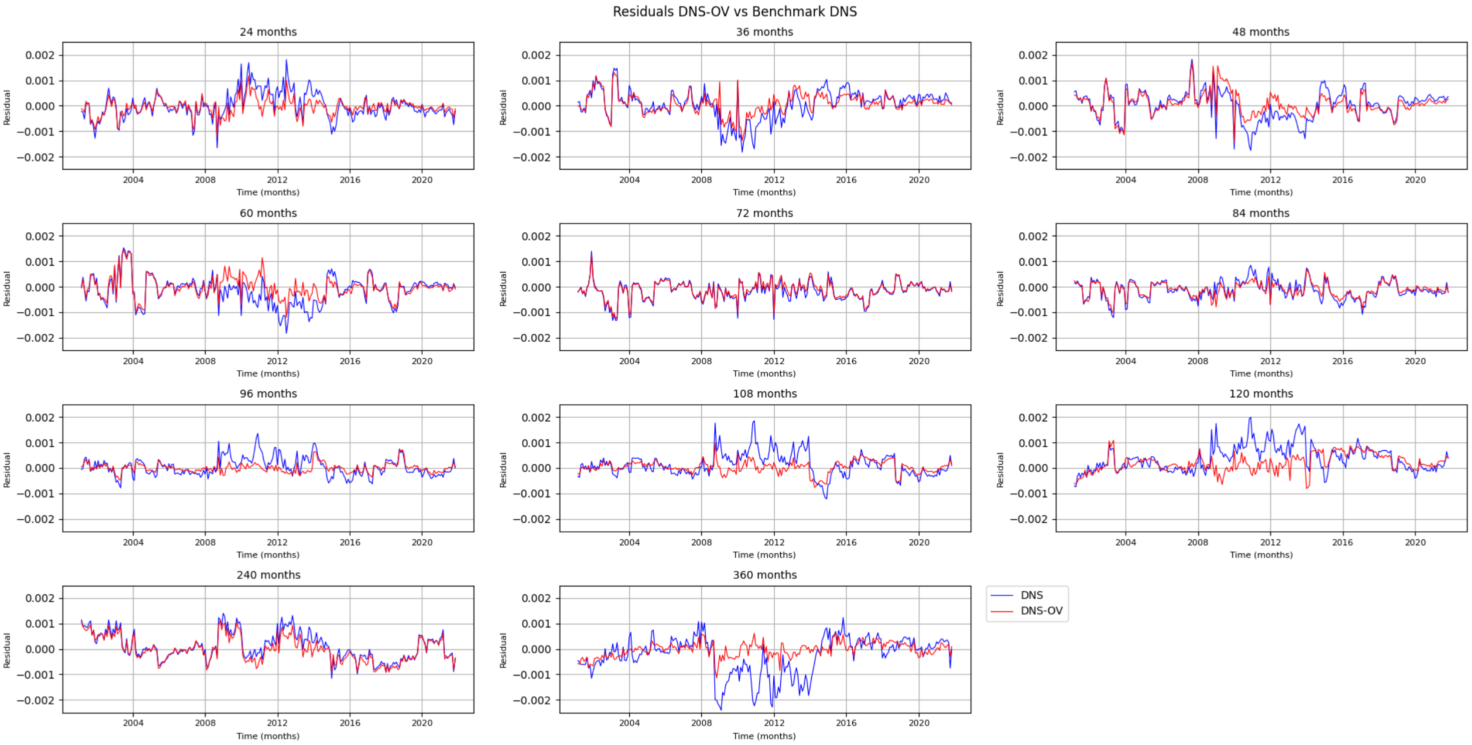


Figure 7.47: Comparison of the residuals of the estimated yields for each maturity between the DNS-OV model and the benchmark DNS model.

Table 7.48: Results of the Ljung-Box test for serial correlation in the residuals of the estimated yields for each maturity with the DNS-OV model.

Maturity τ_i	24	36	48	60	72	84
p value	1.5×10^{-16}	1.9×10^{-25}	3.6×10^{-24}	4.9×10^{-22}	2.7×10^{-21}	5.5×10^{-25}
White noise?	No	No	No	No	No	No
Maturity τ_i	96	108	120	240	360	
p value	1.8×10^{-25}	3.5×10^{-30}	1.4×10^{-31}	1.9×10^{-38}	1.6×10^{-24}	
White noise?	No	No	No	No	No	

7.4.3 Forecasting Analysis

In this subsection we discuss the results of the one-month ahead forecasts of the yield curve in December 2021 and September 2008. The posterior predictive values are simulated 1000 times simultaneously for each maturity. The forecasts are shown in Figure 7.49. First, we notice that the 95% credible regions of the forecast yield curves have some "bulge" for the medium-term maturities, which we have not seen for the previous models. So, it seems that the DNS-OV model has relatively more variability in the medium-term maturities compared with the other maturities. This could be due to the common shock process $\hat{\varepsilon}_t^*$ seemingly affecting the medium-term yields more than the yields of other maturities. Moreover, it is interesting to see that modeling the volatility process explicitly does not result in a better forecast of the shape of the yield curve compared to the benchmark DNS model for the black swan scenario in September 2008. Similarly as the DNS-ARRW model, this could be due to the observation noise σ^+ and the state noise variances q being the same across all maturities resulting in a higher variability for maturities that might have a lower variance if modeled separately. Consequently, we argue that these results give additional reason to model distinct state noise variances and model the observation noise variances grouped for the different segments of the yield curve.

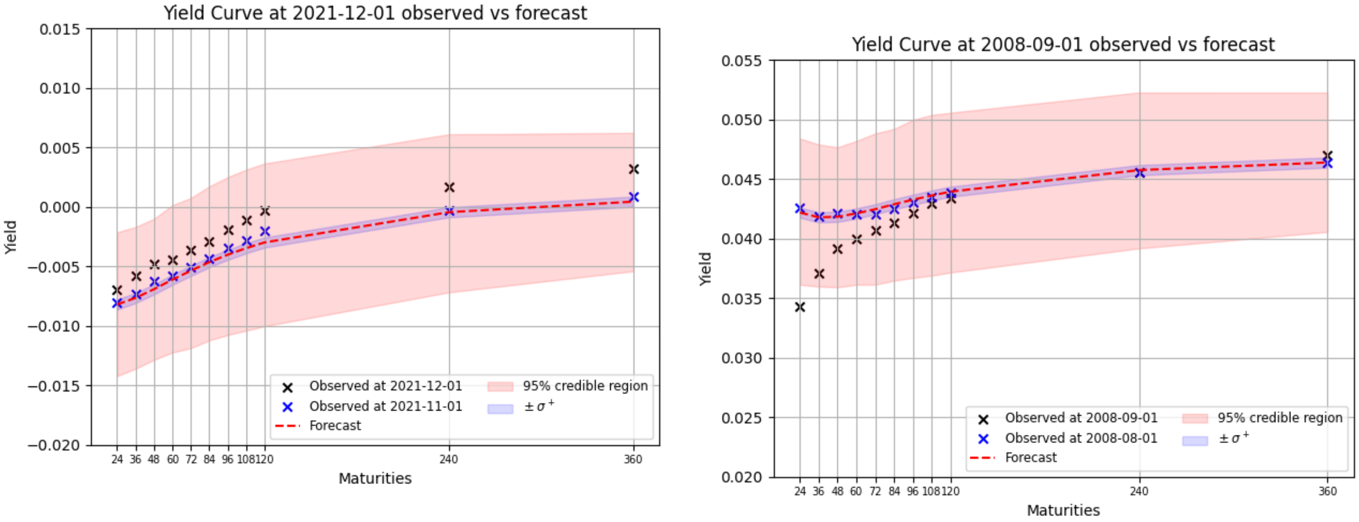


Figure 7.49: The one-month ahead forecast (dashed red line) of December 2021 (left) and September 2008 (right) with 95% credible regions (red surface) and the uncertainty due to the observation noise $\pm\sigma^+$ (blue surface).

7.5 Results of DNS-OVOSN

In this section we discuss the results of the parameter estimation, and the in-sample and forecasting analysis for the last model, the DNS-OVOSN model. This model is a combination of all the previous findings from the results and the literature. Particularly, this is the DNS model with GARCH observation volatility based on Koopman et al. (2010), but with distinct state noise variances and the observation noise variances as well as the volatility loadings grouped by the short (24, 36 months) , medium (48 to 108 months) and long-term (120 to 360 months) segments of the yield curve. Recall that the model is specified in 6.33 and given by

$$
\begin{cases}
\boldsymbol{y}_t = \begin{bmatrix}\Lambda & \Gamma\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_t \\ \varepsilon_t^*\end{bmatrix} + \boldsymbol{\varepsilon}_t^+, & \boldsymbol{\varepsilon}_t^+ \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\Sigma_\varepsilon^S & & \emptyset \\ & \Sigma_\varepsilon^M & \\ \emptyset & & \Sigma_\varepsilon^L\end{bmatrix}\right), \\
\begin{bmatrix}\boldsymbol{\beta}_t \\ \varepsilon_t^*\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu} \\ 0\end{bmatrix} + \begin{bmatrix}\Phi & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_{t-1} \\ \varepsilon_{t-1}^*\end{bmatrix} + \begin{bmatrix}\boldsymbol{\eta}_t \\ \varepsilon_t^*\end{bmatrix}, & \begin{bmatrix}\boldsymbol{\eta}_t \\ \varepsilon_t^*\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\Sigma_\eta & 0 \\ 0 & h_t\end{bmatrix}\right), \\
h_t = \gamma_0 + \gamma_1(\varepsilon_{t-1}^*)^2 + \gamma_2 h_{t-1},
\end{cases}
\tag{7.9}
$$

with parameters $\boldsymbol{\psi} = \{\lambda, \mu_1, \mu_2, \mu_3, \phi_1, \phi_2, \phi_3, \sigma_S, \sigma_M, \sigma_L, q_1, q_2, q_3, \gamma_0, \gamma_1, \gamma_2, \Gamma_S, \Gamma_M, \Gamma_L\}$.

7.5.1 Parameter Estimation

In this subsection we elaborate on the RWM run that results into convergence for the chains of each parameter. In particular, we provide the starting points and scales resulting into convergence in Table 7.50 and Table 7.51. However, we first note that due to slow convergence we have set the starting points for γ_1, γ_2 as 0.7 and 0.2 respectively instead of their MLE values. Then, we also note that this model also has an additional common shock state variable ε_t^* that has a GARCH variance h_t. Since the GARCH process requires an approximation to employ the Kalman filter, we define the initial covariance matrix as $P_0^0 = \text{diag}(p_1^0, p_2^0, p_3^0, p_\varepsilon^0)$ again to have a more accurate initial guess for the log-likelihood computation. Moreover, for this model we have again fixed $\gamma_0 = 0.0001$. Additionally, we note that the high correlation between $\Gamma_S, \Gamma_M, \Gamma_L$ and the fact that they represent values in relation to each other results into bad convergence similar to the DNS-OV model. So, we use the MLE values in Table 7.51 for the volatility loadings again and use the RWM algorithm for the other parameters. Moreover, this RWM run has been run for 60000 iterations and has an average acceptance ratio of $\bar{\alpha} \approx 0.1696$. This indicates a reasonable rate of convergence, especially with the high dimensionality (15 parameters to be estimated) of this model in mind.

Table 7.50: The scales $\boldsymbol{\sigma}_{RWM}$ for each parameter of the DNS-OVOSN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	$\log \sigma_S$
Scale $\sigma_{i,RMW}$	0.00027	0.00018	0.0002	0.00035	0.0045	0.0065	0.0095	0.021
Parameter ψ_i	$\log \sigma_M$	$\log \sigma_L$	$\log q_1$	$\log q_2$	γ_1	γ_2		
Scale $\sigma_{i,RMW}$	0.01	0.015	0.016	0.017	0.029	0.0135		

Then, the results are shown as a trace plot in Figure 7.52 and the histogram of the distributions of each parameter are shown in Figure 7.53. We notice that the chains of the parameters converge relatively well, but the transformed state noise variance parameters $\log q_2, \log q_3$, the GARCH parameters γ_1, γ_2 and to a lesser extent the transformed observation noise variances $\log \sigma_S, \log \sigma_M, \log \sigma_L$ seem to converge a bit slower than the GARCH parameters and the state noise variance counterpart q of the DNS-OV model. As a consequence, the scales of those parameters are relatively large, which result into more uncertain parameter estimation. Part of the slower convergence of γ_1, γ_2 can be due to the volatility loadings being less specific as not every maturity has its own volatility loading, but is grouped together in its respective segment of the yield curve.

Additionally, the slower convergence of the observation noise variance parameters could be due to a possible mismatch between the maturities we have grouped together and the actually related maturities. Subsequently, we have also performed the Geweke diagnostic test with $\tau_A = 0.1, \tau_B = 0.5$ and a significance level of $\alpha = 0.05$, of which the results are presented in Table 7.54. We can see that the test indicates that the means of the first and last segments of the chains of each parameter do not differ significantly. So, considering the trace plot and the Geweke test results, it is reasonable to assume that the chains of the parameters have converged.

Subsequently, the MAPE is used for the in-sample analysis again and is provided in Table 7.55. Interestingly, the GARCH parameters γ_1, γ_2 for the DNS-OVOSN model are similar to the values of the GARCH parameters for the DNS-OV model as both are close to 0.8 and 0.01 respectively. Moreover, we notice that the observation noise standard deviation σ_L^+ for the the long-term maturities is more than two times larger than the observation noise standard deviations of the medium-term and short-term maturity yields σ_M^+ and σ_S^+ respectively. This could indicate that the DNS-OVOSN model has more difficulty in estimating the long-term maturities. Additionally, we also notice that the state noise standard deviations q_1, q_2 and q_3 are similar to the state noise standard deviations of the DNS-SN model, which are 0.00203, 0.0035 and 0.00501 for $\beta_{1,t}, \beta_{2,t}, \beta_{3,t}$ respectively for the latter model. Finally, we can see that the BIC and AIC values for this model with the MAPE values are in-between the previous models as its goodness-of-fit is better than the benchmark DNS (BIC: -30616, AIC: -30648) and DNS-SN model (BIC: -30723, AIC: -30762), but worse than the DNS-ARRW model (BIC: -32815, AIC: -32878) and DNS-OV model (BIC: -31573, AIC: -31654). The large difference with the DNS-OV model is somewhat remarkable, since the DNS-OVOSN has more flexibility in the noise processes, but has more restricted volatility loadings. So, one could argue that the volatility loadings have a greater benefit for the in-sample fit compared to distinct noise variance.

Table 7.51: The MLE values as RWM starting points $\boldsymbol{\psi}^{(0)}$ of the parameters approximated by the L-BFGS-B minimizer and the corresponding log-likelihood value for the DNS-OVOSN model.

Parameter ψ_i	λ	$\hat{\beta}_1^0$	$\hat{\beta}_2^0$	$\hat{\beta}_3^0$	$\hat{\varepsilon}^{*0}$	p_1^0	p_2^0	p_3^0	p_ε^0
Bounds	$(10^{-4}$,$\infty)$	$(-\infty$,$\infty)$	$(-\infty$,$\infty)$	$(-\infty$,$\infty)$	$(-\infty$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$
Initial guess	0.1	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0
MLE value	0.05046	3×10^{-5}	0.0	-1×10^{-5}	0.0	0.99959	0.99956	0.99964	1.0
Parameter ψ_i	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	σ_S^+	σ_M^+	σ_L^+
Bounds	$(-\infty$,$\infty)$	$(-\infty$,$\infty)$	$(-\infty$,$\infty)$	$(-1$,$1)$	$(-1$,$1)$	$(-1$,$1)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$
Initial guess	0.0	0.0	0.0	0.99	0.99	0.99	0.01	0.01	0.01
MLE value	-0.00014	-0.00057	-0.00174	0.99696	0.96545	0.94801	0.00036	0.00042	0.00097
Parameter ψ_i	q_1	q_2	q_3	γ_1	γ_2	Γ_S	Γ_M	Γ_L	
Bounds	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$\infty)$	$(10^{-4}$,$1)$	$(10^{-4}$,$1)$	$(10^{-5}$,$\infty)$	$(10^{-5}$,$\infty)$	$(10^{-5}$,$\infty)$	
Initial guess	0.01	0.01	0.01	0.45	0.45	0.1	0.1	0.1	
MLE value	0.00172	0.00302	0.00511	0.42753	0.40834	0.02793	0.0496	0.02986	
Log-likelihood	15574								

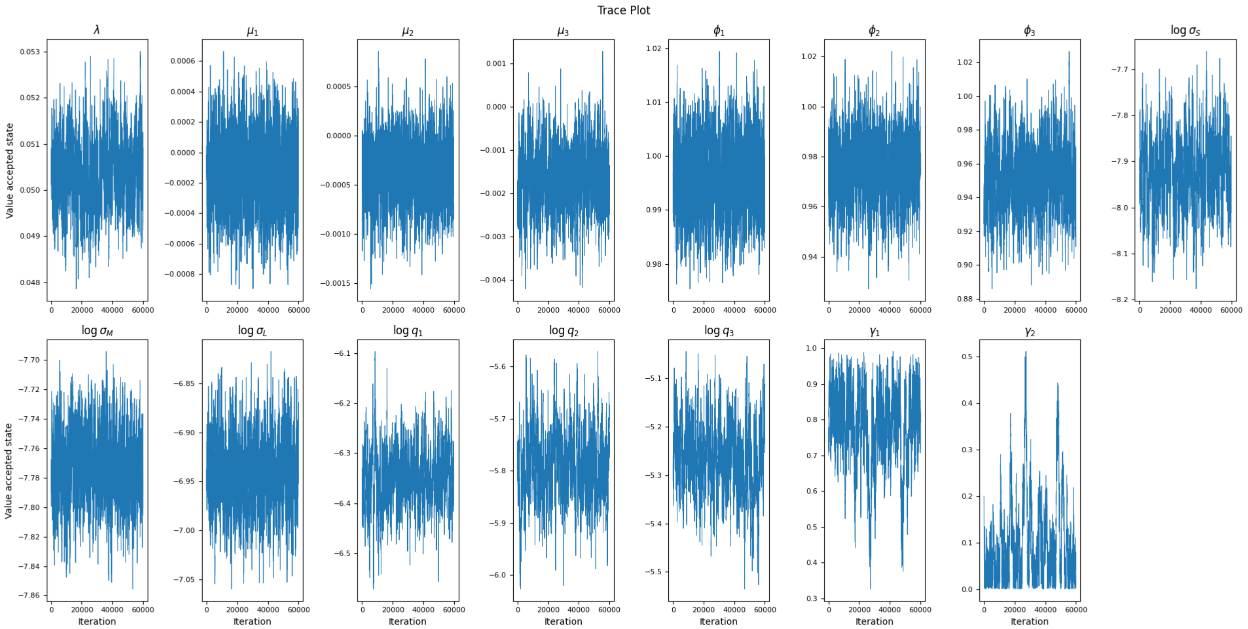


Figure 7.52: Trace plot of the chains for each parameter of the DNS-OVOSN model.

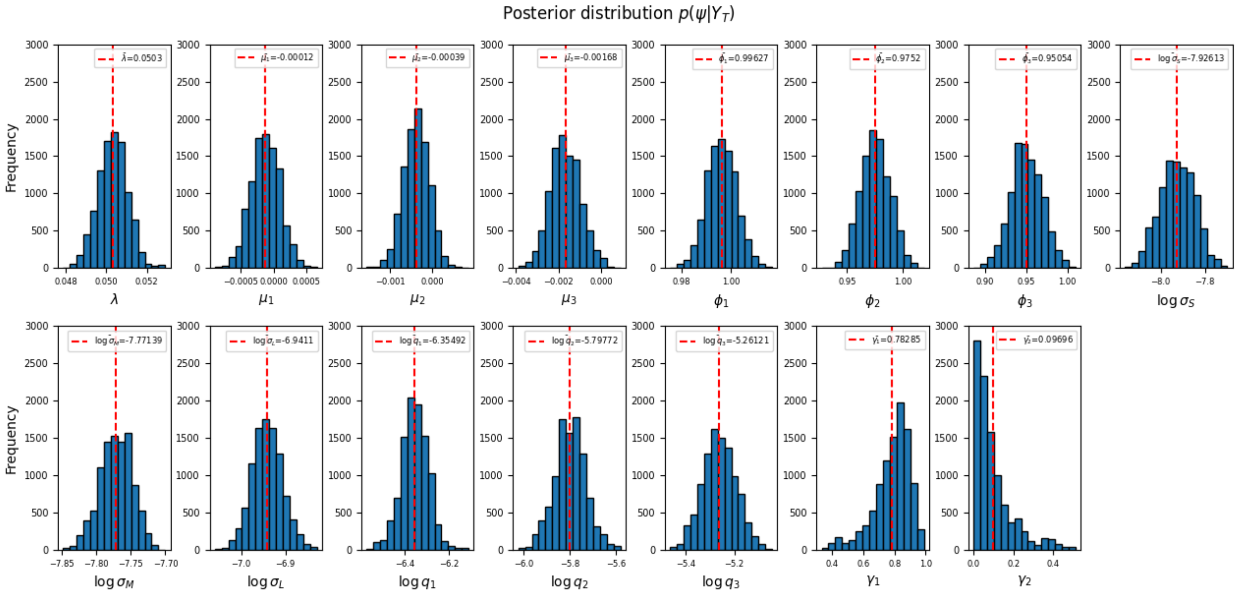


Figure 7.53: Histogram with the mean of the posterior distribution for each parameter of the DNS-OVOSN model.

Table 7.54: The results of the Geweke diagnostic test for the chains of each parameter of the DNS-OVOSN model.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	σ_S
Statistic $\lvert G \rvert$	0.14260	0.00862	0.02018	0.19667	0.00349	0.07271	0.20770	0.10458
Means differ significantly?	No	No	No	No	No	No	No	No
Parameter ψ_i	σ_M	σ_L	q_1	q_2	q_3	γ_1	γ_2	
Statistic $\lvert G \rvert$	0.02203	0.00759	0.32864	0.29421	0.37467	0.39821	0.51710	
Means differ significantly?	No	No	No	No	No	No	No	

Table 7.55: The MAPE values $\hat{\boldsymbol{\psi}}^{MAPE}$ for the DNS-OVOSN model and the corresponding log-likelihood (LL), BIC and AIC values.

Parameter ψ_i	λ	μ_1	μ_2	μ_3	ϕ_1	ϕ_2	ϕ_3	σ_S
MAPE $\hat{\psi}_i^{MAPE}$	0.05012	-0.00017	-0.00049	-0.00181	0.99414	0.97066	0.94294	0.00036
Parameter ψ_i	σ_M	σ_L	q_1	q_2	q_3	γ_1	γ_2	
MAPE $\hat{\psi}_i^{MAPE}$	0.00043	0.00095	0.00174	0.00314	0.00497	0.87940	0.01600	
Model fit measure	LL	BIC	AIC					
Value	15576	-31047	-31114					

7.5.2 In-Sample Analysis

In this subsection we discuss the in-sample results of the DNS-OVOSN model. Since this model also has a volatility process, we will compare the estimated volatility of the DNS-OVOSN model with the DNS-OV model.

Estimated State Variables and Yields

We use the Kalman filter in a similar way as we have used for the DNS-OV model to estimate the state variables $(\hat{\boldsymbol{\beta}}_t, \hat{\varepsilon}_t^*)$. We note that we have used the MLE values of the initial state and initial standard deviations for the Kalman filter as in Table 7.51 and the MAPE values of the model parameters as in Table 7.55 for this analysis.

The results of the state and subsequent yield estimation are presented in Figure 7.56. The original Nelson-Siegel state variables are similar to the estimates of the benchmark DNS model. Although, we can see that the DNS-OVOSN model also seems to affect the curvature $\hat{\beta}_{3,t}$ the most. Moreover, the common shock process $\hat{\varepsilon}_t^*$ shows interesting behaviour. First, the common shock process exhibits a quite large negative shock in 2003 and around 2015, which we have not seen for the DNS-OV model. In addition, the process $\hat{\varepsilon}_t^*$ shows more smaller negative shocks along the entire period of 2001 to 2021, whereas the common shock process for the DNS-OV model generally showed positive shocks. Secondly, both models DNS-OVOSN and DNS-OV show extreme positive peaks between 2008 and 2012, but the DNS-OVOSN model shows less persistent behaviour of $\hat{\varepsilon}_t^*$.

Interestingly, the common shock of DNS-OVOSN seems to model the volatility following both the dotcom crash (2000-2002) and the financial crisis (2008-2009) as opposed to the DNS-OV model that has only estimated high volatility around 2008 to 2012. The negative shock coincides with a short period, in which the yields decrease significantly while the medium-term yields stay closer to the short-term yields. On the contrary, the positive shocks seem to coincide with a period that also sees significant decreases of yields, but with medium-term yields that seem further away from both short-term and long-term yields. In order to have a better grasp of these yield curves we show the yield curves on three dates in a similar way as the DNS-ARRW and DNS-OV models in Figure 7.57. Recall that these yield curves are based on the state variables $\hat{\boldsymbol{\beta}}_t$, so without the common shock process. However, in order to show the effect of $\hat{\varepsilon}^*_t$ on the estimated yields, the estimates based on all state variables are shown as well. The three considered dates are May 2003 (negative peak of $\hat{\varepsilon}^*_t$), May 2009 (positive peak of $\hat{\varepsilon}^*_t$) and January 2020, which sees a relatively low value of $\hat{\varepsilon}^*_t$. Then, we can see that the observed yields in May 2003 have short-term and medium-term segments that are increasing more linearly and have ca. 200 bps between the maturities of 24 and 120 months, while the gap between the maturities of 120 and 360 months is ca. 90 bps. In contrast, the observed yields in May 2009 increase faster for the short-term maturities before increasing slowly for the long-term maturities. Specifically, the gap between the maturities of 24 and 120 months is ca. 300 bps, whereas the gap between the maturities of 120 and 360 months is around 50 bps. So, it seems that $\hat{\varepsilon}^*_t$ models some underlying process affecting the volatility not only in increases or decreases of the medium-term yields like the DNS-OV model, but it also seems to affect the "steepening" or "flattening" of the yield curve.

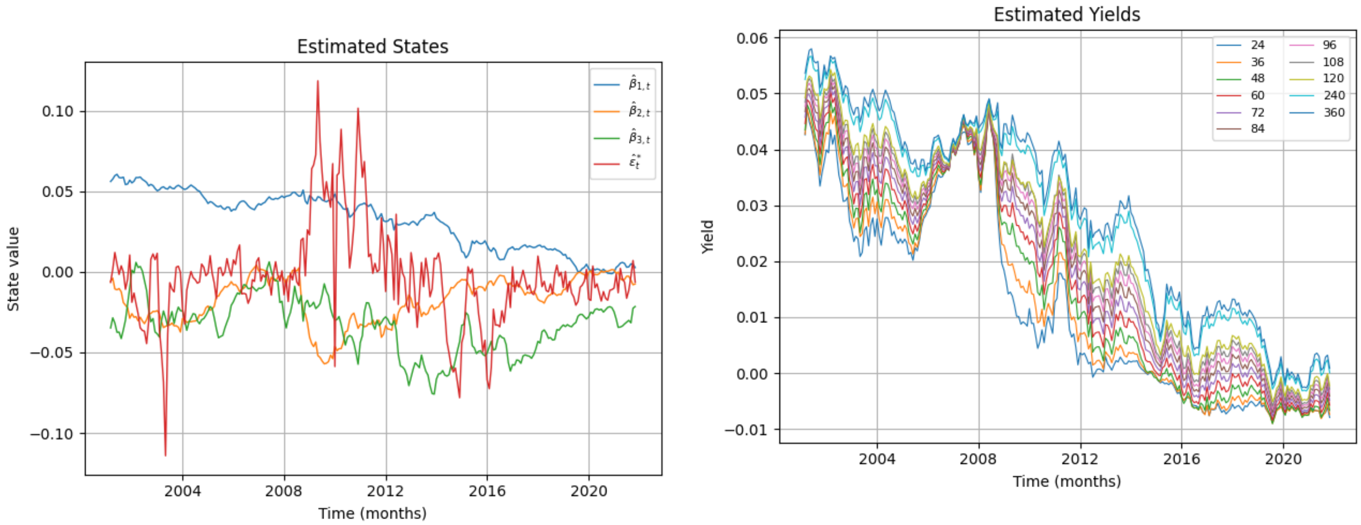


Figure 7.56: The estimated state variables $\hat{\beta}_{1,t}, \hat{\beta}_{2,t}, \hat{\beta}_{3,t}$ and $\hat{\varepsilon}^*_t$ for $t = 1, \ldots, 249$ obtained by the Kalman filter and the estimated yields $\hat{\boldsymbol{y}}_t = \Lambda\hat{\boldsymbol{\beta}}_t + \Gamma\hat{\varepsilon}^*_t$ with the DNS-OVOSN model.

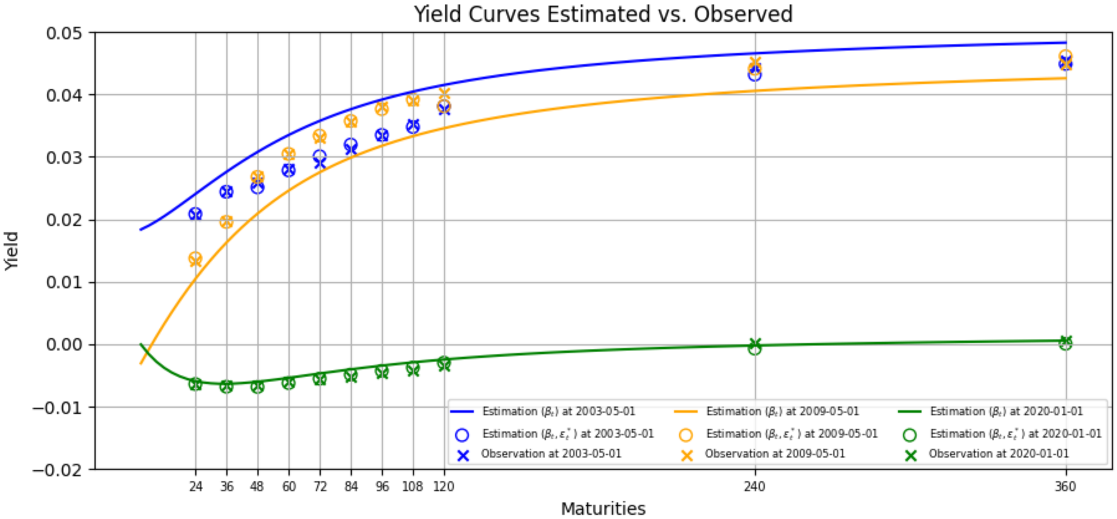


Figure 7.57: The observed (crosses) and estimated yield curves based on only the state variables $\hat{\boldsymbol{\beta}}_t$ (smooth curves) and based on all states (circles) in May 2003 (blue), May 2009 (orange) and January 2020 (green). The estimated yields are based on the DNS-OVOSN state variables inside the parentheses in the legend.

Then, recall that the volatility process h_t is the variance of the common shock process $\hat{\varepsilon}_t^*$. The volatility loadings determine to what extent this volatility is translated to each specific maturity. For the DNS-OVOSN model we have assumed that each segment of the yield curve has its own volatility loading Γ_S, Γ_M and Γ_L for the short, medium and long-term maturities respectively. In Figure 7.58 the volatility loadings of the DNS-OVOSN model are compared with the volatility loadings of the DNS-OV model. Recall that restricting the number of volatility loadings had significant impact on the in-sample fit of the DNS-OVOSN model, but it allows for less parameters and consequently for less overfitting. We can see that the volatility loadings for the DNS-OVOSN model are lower than the loadings of the DNS-OV. This might be due to the modeling of distinct state noise variances and observation noise variances. Specifically, if the noise processes are more "tailored" to each state or segment of the yield curve, then we can expect that the noise process can explain more of the random errors between the estimated and observed yields. This could result into less need for the volatility for each maturity to explain the random errors, resulting into smaller volatility loadings. Additionally, the volatility loadings of the DNS-OVOSN model seem to follow a similar pattern as the loadings of the DNS-OV model, in which the loadings increase and decrease as the maturity increases. Subsequently, the three volatility loadings of DNS-OVOSN result into three types of volatility. In particular, in Figure 7.59 the short, medium and long-term volatility processes $\Gamma_S^2 h_t, \Gamma_M^2 h_t, \Gamma_L^2 h_t$ are shown together with the volatility loadings of the DNS-OV model for the maturities of 24, 72 and 360 months. Similar to the common shock process, we see that there is quite some volatility around 2003, 2009 and 2015 for the DNS-OVOSN model, whereas the volatility of the DNS-OV model concentrates mainly around 2008 to 2012.

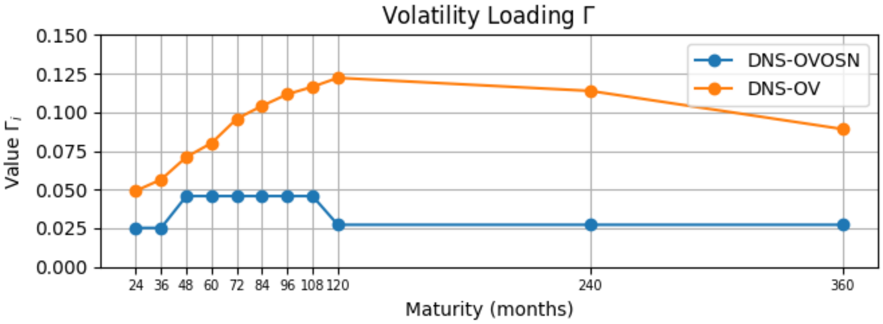


Figure 7.58: The values of the DNS-OVOSN volatility loadings $\Gamma_S, \Gamma_M, \Gamma_L$ (blue) compared with the DNS-OV volatility loadings $\Gamma_1, \ldots, \Gamma_{11}$ (orange) for maturities $24, \ldots, 360$ months respectively.

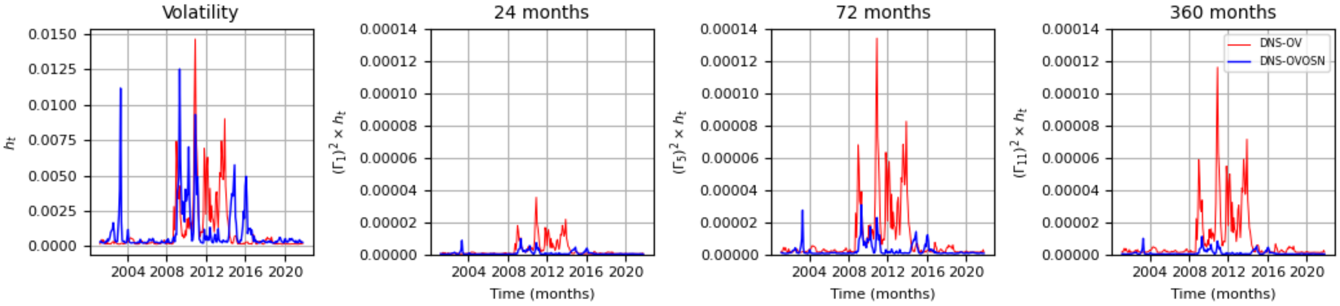


Figure 7.59: Comparison between the volatility process h_t of the DNS-OVOSN model (blue) and the DNS-OV model (red) together with the DNS-OVOSN volatilities $\Gamma_S^2 h_t, \Gamma_M^2 h_t, \Gamma_L^2 h_t$ (blue) and the DNS-OV volatilities $\Gamma_1^2 h_t, \Gamma_5^2 h_t, \Gamma_{11}^2 h_t$ (red) corresponding to maturities of 24, 72 and 360 months.

Residuals Analysis

The residuals of the yields estimated by the DNS-OVOSN model are compared with the benchmark DNS model for each maturity in Figure 7.60. Similar to the previous models we have assumed that the observation noise follows a Normally distributed white noise $\varepsilon_t^* \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon^+)$, so we would expect that the residuals show no serial correlation. We can see that the DNS-OVOSN model seems to estimate the short-term maturities of 24 and 36 months quite well and even show improvement compared to the benchmark DNS model. Subsequently, the medium-term maturities of 48 to 108 seem to be comparable to the benchmark DNS model and do not show the same improvements as the DNS-OV model. Finally, the residuals of the long-term maturities of 120 to 360 months seem to perform comparable or to some extent worse than the benchmark DNS model. In particular, the DNS-OVOSN model seems to have difficulty with estimating the yields of the longest-term maturity of 360 months. We have seen that grouping the volatility loadings by segments of the yield curve resulted into less in-sample performance based on the BIC and AIC values compared with the DNS-OV model. So, the worse performance on the longest-term maturity could be caused by a suboptimal grouping of the 360 months maturity with the 240 and 120 months maturities as one long-term segment with the same observation noise variance and volatility loading.

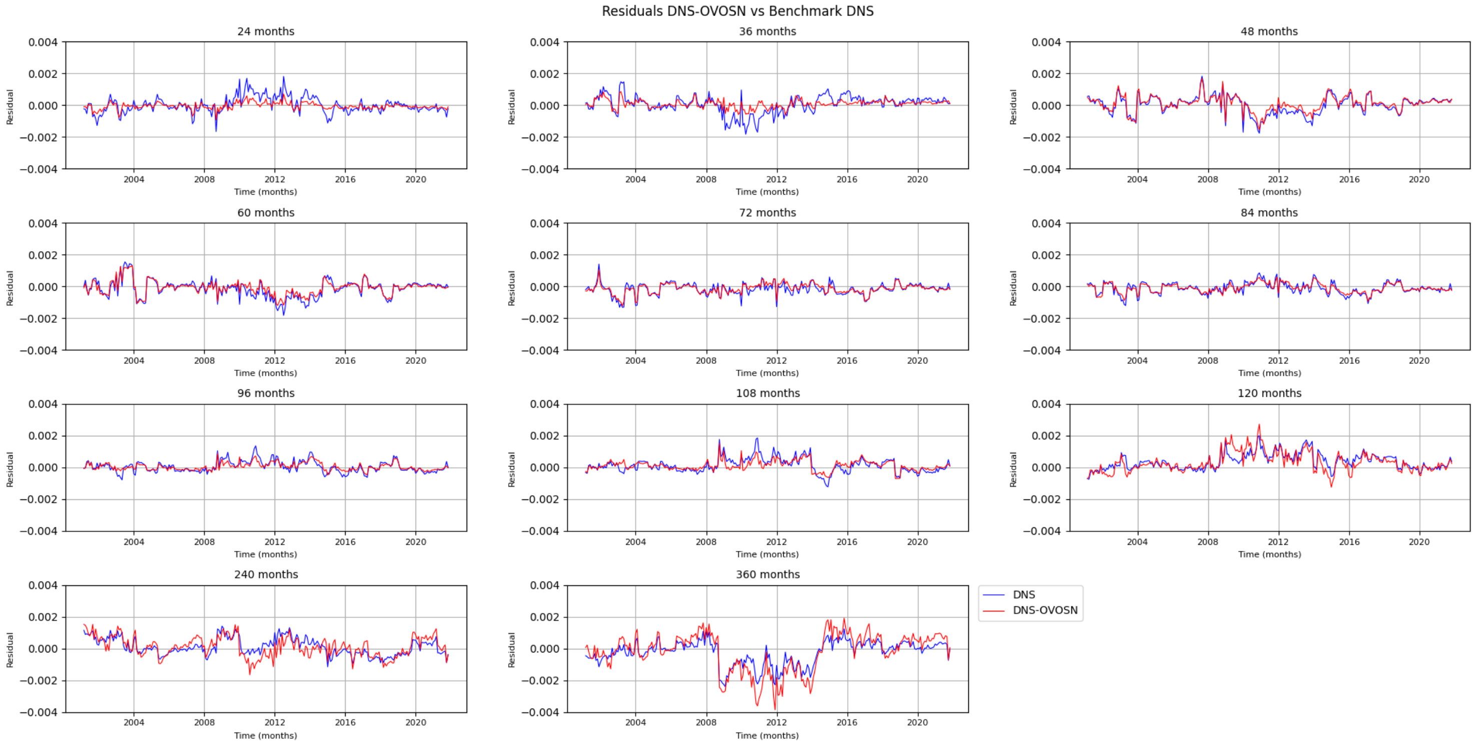


Figure 7.60: Comparison of the residuals of the estimated yields for each maturity between the DNS-OVOSN model and the benchmark DNS model.

Then, we perform a Ljung-Box test with significance level $\alpha = 0.05$ and lag $h = 1$ to test whether the residuals resemble a white noise process. The results are provided in Table 7.61. We note that the p values for the residuals of every maturity are significantly low, which indicates that the residuals of the yields estimated by the DNS-OVOSN model show significant correlation. So, the residuals are not likely a white noise process.

Table 7.61: Results of the Ljung-Box test for serial correlation in the residuals of the estimated yields for each maturity with the DNS-OVOSN model.

Maturity τ_i	24	36	48	60	72	84
p value	3.0×10^{-18}	6.1×10^{-21}	8.7×10^{-28}	4.0×10^{-30}	1.6×10^{-26}	1.4×10^{-32}
White noise?	No	No	No	No	No	No
Maturity τ_i	96	108	120	240	360	
p value	3.9×10^{-28}	4.1×10^{-27}	3.7×10^{-35}	3.7×10^{-32}	5.6×10^{-45}	
White noise?	No	No	No	No	No	

7.5.3 Forecasting Analysis

In this subsection we discuss the results of the one-month ahead forecasts of the yield curve in December 2021 and September 2008. Again, for each maturity the posterior predictive values are simultaneously simulated 1000 times. Notice that the 95% credible regions have a much smaller "bulge" around the medium-term maturities compared with the DNS-OV forecasts. So, it seems that the variability of the medium-term yields is not significantly affected. This is in line with our in-sample findings that the common shock process $\hat{\varepsilon}_t^*$ seems to affect the medium-term yields more in the "steepening" or "flattening" of the yield curve as opposed to the common shock process of the DNS-OV model that seemingly affects the yields across the maturities. Moreover, the increase of the yields and the shape of the yield curve in December 2021 seem to be reasonably forecast as they are inside the credible region and close to the mean of the forecasts. On the other hand, the forecast of the shortest-term maturity of 24 months seems to be further away from the observed yield than was the case for the DNS-OV model, but not by a large amount. In particular, the absolute difference between the lower bound of the 95% credible region of the DNS-OVOSN for the 24 month maturity is 21 bps compared to the DNS-OV model with 18 bps. In summary, the DNS-OVOSN seems to have a comparable one-month ahead forecasting performance even though it has more restricted volatility loadings.

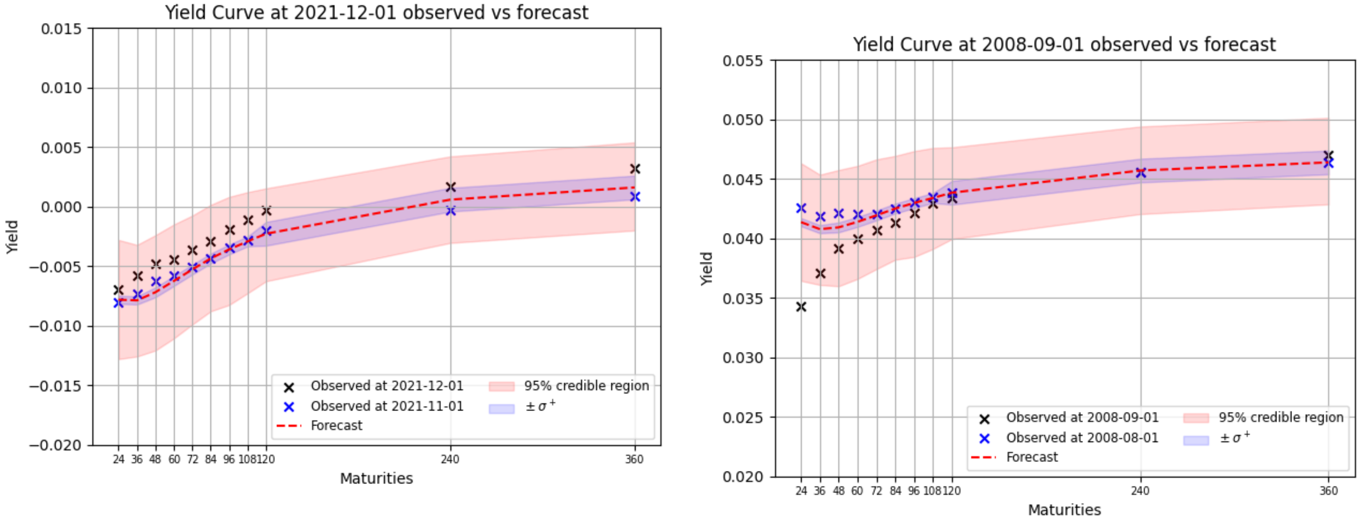


Figure 7.62: The one-month ahead forecast (dashed red line) of December 2021 (left) and September 2008 (right) with 95% credible regions (red surface) and the uncertainty due to the observation noise $\pm\sigma^+$ (blue surface).

7.6 Comparison with Current Model and Method

In this section we compare the explored models with the current model and method that is employed by the DSTA, as described in Subsection 6.2.1. We will compare the current model and method with our explored models and the Bayesian approach for in-sample performance and forecasting performance.

7.6.1 In-Sample Comparison

As aforementioned, in this subsection we compare all the explored model with the current model. Notice that for the current model we have no log-likelihood approximation as is the case for the explored state-space models. So, we cannot compare the current model based on goodness-of-fit measures as the BIC or the AIC. Consequently, we will use the so-called *root mean squared error*, or RMSE in short. We will compute the RMSE for each maturity for every model, defined as

$$\text{RMSE}(\tau_i) = \sqrt{\sum_{t=1}^{T} \frac{\left(\hat{y}_t(\tau_i) - y_t(\tau_i)\right)^2}{T}}, \tag{7.10}$$

where $\hat{y}_t(\tau_i)$ is the estimated yield at time t for maturity τ_i, $y_t(\tau_i)$ is the corresponding observation of the yield at time t and $T = 249$. Additionally, we also compute the total RMSE of every model, which is defined as

$$\text{Total RMSE} = \sum_{i=1}^{M} \text{RMSE}(\tau_i), \tag{7.11}$$

where M denotes the total number of maturities ($M = 11$ in our case). In Table 7.63 the RMSE values of the models are given for each maturity and in the last row the total RMSE is provided. These in-sample RMSE values are also visualized in Figure 7.64 for each maturity. Together the values give a indication of how much the in-sample yield estimations deviate from the observed yields. We have divided the linear yield curve models without volatility and the nonlinear yield curve model with volatility modeling.

First, we notice that the DNS-ARRW model outperforms all the other models with regard to fitting in-sample yield data. In the one-step ahead forecasting analysis of the DNS-ARRW model in Subsection 7.3.3 we have seen that the 95% credible region of the forecast was quite wide compared to the other models. This can be put into two perspectives. On the one hand, this could mean that the DNS-ARRW model perhaps is overfitting the in-sample yield data, which results into less accurate forecasts of out-of-sample yields. On the other hand, it could also indicate that the DNS-ARRW can potentially forecast the 95% credible region upper bound (worst-case scenario) well if we forecast multiple steps ahead. This is discussed into more detail in the next section. Then, overall we can see that the volatility model DNS-OV outperforms every other model except for the DNS-ARRW model and the DNS-OVOSN model for three shorter-term maturities. Additionally, the DNS-OVOSN seems to have a good performance on the short-term and medium-term maturities, but the in-sample performance on the long-term maturities drops significantly. Moreover, the performance of the current model is comparable with the benchmark DNS and DNS-SN models. This means that the current model performs relatively good compared with its closest state-space counterparts. Moreover, looking at the more complex extensions (DNS-ARRW, DNS-OV), a lot of additional performance can be gained with model extensions for at least the in-sample estimation of the yields. Furthermore, the DNS-OV and the DNS-ARRW models are the only models that consistently outperform the current model in fitting the in-sample yield data across all maturities. Additionally, the DNS-OVOSN model outperforms the current model as well for the maturities of 24 to 108 months. However, as we have seen in the in-sample analysis of the DNS-OVOSN model (Subsection 7.5.2) this could be the result of grouping the volatility loadings of those maturities too restrictively.

In summary, the current model and method seem to perform quite well compared to the benchmark DNS and DNS-SN models. However, the DNS-ARRW and the DNS-OV models outperform the current model and method for every maturity and to a lesser extent the DNS-OVOSN model for only the short-term and medium-term maturities.

Table 7.63: Comparison of the in-sample RMSE of each model for the different maturities and the total RMSE for each model. The best performance inside the type of models is in **bold**.

	Linear Yield Curve models				Volatility models	
Maturity	Current DNS	Benchmark DNS	DNS-SN	DNS-ARRW	DNS-OV	DNS-OVOSN
24	0.000387	0.000507	0.000451	**0.000066**	0.000331	**0.000229**
36	0.000587	0.000576	0.000579	**0.000077**	0.000423	**0.000283**
48	0.000561	0.000565	0.000561	**0.000090**	**0.000459**	0.000505
60	0.000514	0.000574	0.000549	**0.000083**	**0.000438**	0.000464
72	0.000348	0.000411	0.000384	**0.000063**	0.000367	**0.000342**
84	0.000364	0.000378	0.000369	**0.000051**	**0.000304**	0.000310
96	0.000340	0.000359	0.000355	**0.000043**	**0.000185**	0.000250
108	0.000446	0.000467	0.000466	**0.000059**	**0.000249**	0.000357
120	0.000582	0.000621	0.000614	**0.000066**	**0.000401**	0.000696
240	0.000506	0.000549	0.000526	**0.000050**	**0.000489**	0.000620
360	0.000769	0.000791	0.000778	**0.000057**	**0.000284**	0.001166
Total	0.005406	0.005799	0.005632	**0.000707**	**0.003931**	0.005223

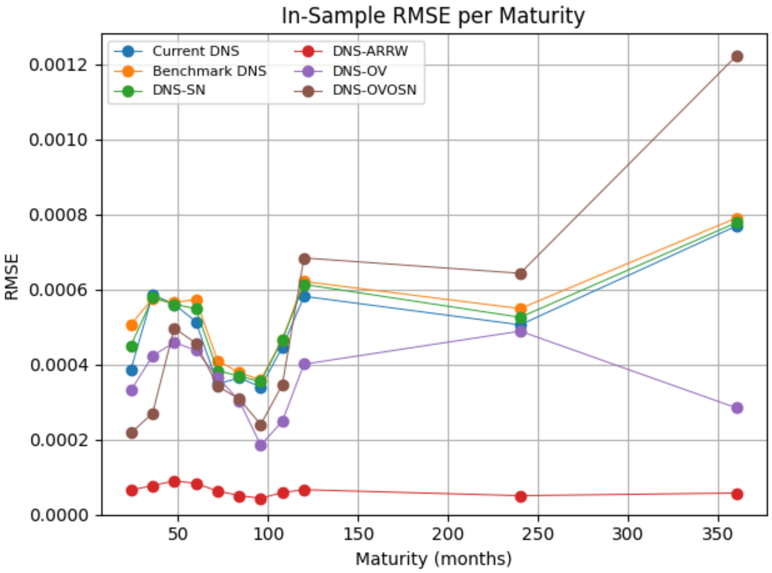


Figure 7.64: Visualization of the in-sample RMSE of each model for the different maturities.

7.6.2 Twelve-Months Ahead Forecast Comparison

In this subsection we compare the forecasting performance of the current model, denoted as Current DNS, with the explored models similar to the in-sample comparison. In particular, the 12-months ahead forecasts from December 2021 to November 2022 based on 1000 simulation paths are compared. Recall that the aim of this thesis is to improve the forecasts of bond yields. Consequently, we compare the worst-case forecasts that are defined as the upper bound of the 95% credible region of the forecasting simulations. A drawback of this method is that it does not take into account the direction of the mean forecasts or the lower bound of the credible regions. However, one can argue that if the worst-case scenario of the yield forecasts is not close to the actual yields, then a model or a method loses significance in practice as is the case with the current model and method. Subsequently, the RMSE of the worst-case forecasts for each model with the Bayesian approach is compared with the worst-case forecasts of the current model and method. The RMSE for each maturity as well as the total RMSE is provided in Table 7.65. The RMSE values for each model and maturity are also visualized in Figure 7.66. Furthermore, the 12-months ahead forecasts of the yields for every maturity are shown in Figure 7.68, 7.67 and 7.69 for the benchmark DNS model, the DNS-ARRW model and the DNS-OVOSN model respectively. The additional 12-months ahead forecasts of the other models can be found in Appendix A.2. The reason for only showing those three selected models is that the benchmark DNS model serves as a baseline for using a Bayesian forecasting approach. In addition, the forecast of the DNS-ARRW model is shown since it consistently shows the best performance on the measures that we have considered for the in-sample and forecasting comparison. Moreover, we also show the forecast of the DNS-OVOSN model since this model combines all of the earlier findings of the explored models.

In Table 7.65 we can see that the DNS-ARRW model outperforms all the other models for each maturity except for the benchmark DNS model for the shortest-term maturity of 24 months. Interestingly, looking at Figure 7.66 the DNS-ARRW model seems to have more accurate forecasts for the longer-term maturities as opposed to the other models that tend to have a quite stable performance across maturities (Benchmark DNS, DNS-OV) or better medium-term forecasts (Current DNS, DNS-SN, DNS-OVOSN). However, looking at the corresponding 12-months ahead forecast of the DNS-ARRW model in Figure 7.67, this seems to be mainly due to the fact that the RMSE does not consider the difference between an observed yield being inside or outside the credible region. Since the DNS-ARRW has wider credible regions for all future months, this results into a little higher RMSE for the first few months while the RMSE decreases as the model approaches the observed yields further away in time significantly. It seems likely that the additional variability in the simulated yields stems from the relatively large variance in the posterior samples of the AR(1) parameters $\alpha_2, \ldots, \alpha_9$ similar to the one-month ahead forecasts. For a more detailed discussion on this uncertainty we refer to Subsection 7.3.3. Furthermore, the relatively wide credible regions can indicate an inaccurate forecast as there seem to be as many simulation paths steeply decreasing as increasing. On the contrary, if we would not have known the actual development of the yields in November 2021, then after one year in November 2022 this worst-case forecast would have been the closest to the actual yields compared with the other models.

Table 7.65: Comparison of the RMSE for the worst-case (upper bound of the 95% credible region) forecasts of each model for the different maturities and the total RMSE for each model. The best performance inside the type of models is in **bold**.

	Linear Yield Curve models				Volatility models	
Maturity	Current DNS	Benchmark DNS	DNS-SN	DNS-ARRW	DNS-OV	DNS-OVOSN
24	0.007795	**0.005448**	0.007268	0.006064	**0.006732**	0.008801
36	0.008559	0.005495	0.007815	**0.004969**	**0.006741**	0.009801
48	0.008937	0.005486	0.008042	**0.004569**	**0.006554**	0.009748
60	0.009407	0.005795	0.008387	**0.004462**	**0.006798**	0.009912
72	0.009603	0.005752	0.008499	**0.004291**	**0.006777**	0.009829
84	0.009675	0.005714	0.008556	**0.004147**	**0.006655**	0.009748
96	0.009568	0.005612	0.008511	**0.003987**	**0.006323**	0.009430
108	0.009635	0.005573	0.008655	**0.003993**	**0.006258**	0.009274
120	0.010206	0.005962	0.009150	**0.004131**	**0.006588**	0.009841
240	0.009641	0.005466	0.008770	**0.004312**	**0.006058**	0.008903
360	0.008052	0.004147	0.007326	**0.003520**	**0.004733**	0.007194
Total	0.101079	0.060451	0.090978	**0.048446**	**0.070210**	0.102481

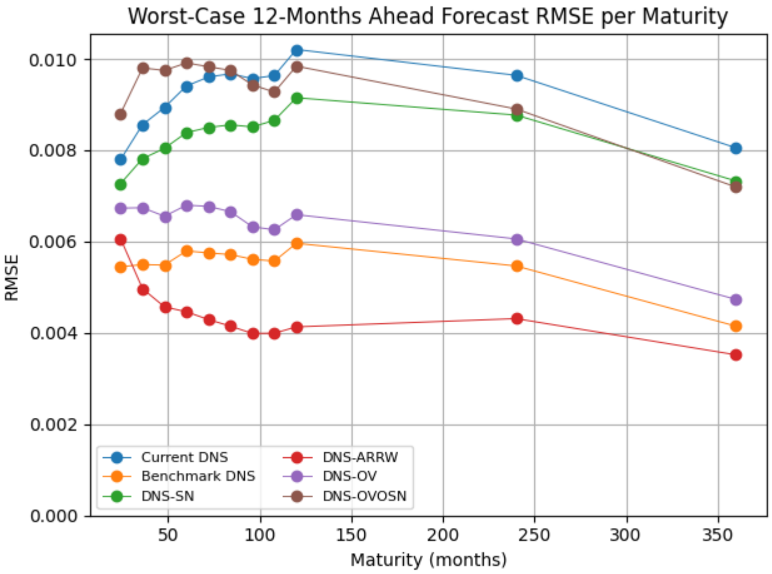


Figure 7.66: Visualization of the worst-case 12-months ahead forecast (upper bound of the 95% credible region) RMSE of each model for the different maturities.

Then, in Figure 7.68 the comparison between the current model and the benchmark DNS model is shown. We can see that in general the current model has narrower credible regions and does not seem to capture the variability in the higher yields compared to the benchmark DNS model. This is in line with our expectation that the current model does not seem to model the upward trend of the yields of the period after November 2021. Although, the benchmark DNS model has credible regions that are quite wide, especially for the longer-term maturity forecasts. The most significant difference between the two models is mainly the Bayesian approach to estimating the parameters as opposed to the MLE-based parameter estimation of the current model. So, it seems that the additionally modeled uncertainty of the parameters results into better quantification of the uncertainty especially in allowing more variability in increasing yield forecasts.

Furthermore, it is also interesting to see that the DNS-OVOSN model has the worst performance based on the RMSE of the worst-case forecast. The performance of this model is the closest to the current model. It seems that especially for the shorter-term maturities (24, 36 and 48 months) the current model has better worst-case forecasts, whereas the DNS-OVOSN model has improved forecasts on mainly the longer-term maturities (240 and 360 months). Then, looking at the comparison of the forecasts of both models in Figure 7.69, we can see that for most of the maturities the forecasts are very close. However, the current model seems to still simulate more yield paths that decrease compared to the DNS-OVOSN model. Specifically, for the maturities of 24 to 48 months we can see that the credible region of the current model forecasts are concentrated to yields that are a little lower than the credible region of the DNS-OVOSN forecasts. Moreover, for the long-term maturities of 240 and 360 months the current model seems to have credible regions that are overall a little lower than the credible region of the DNS-OVOSN model.

In summary, the forecasts of the current model seem to allow for more variability in the lower yields and to some extent the benchmark DNS model as well. In contrast, the forecasts of the DNS-OVOSN model are comparable with the current model, but the DNS-OVOSN forecasts seem to allow for more upward yield simulations whereas the current model seems to simulate yields that are more concentrated in lower yields. Overall it seems that the forecasts of the DNS-ARRW model seem to approach the actually observed yields the closest. So, we consider the DNS-ARRW model to have the best forecasting power based on the ability to simulate the variability of the yields.

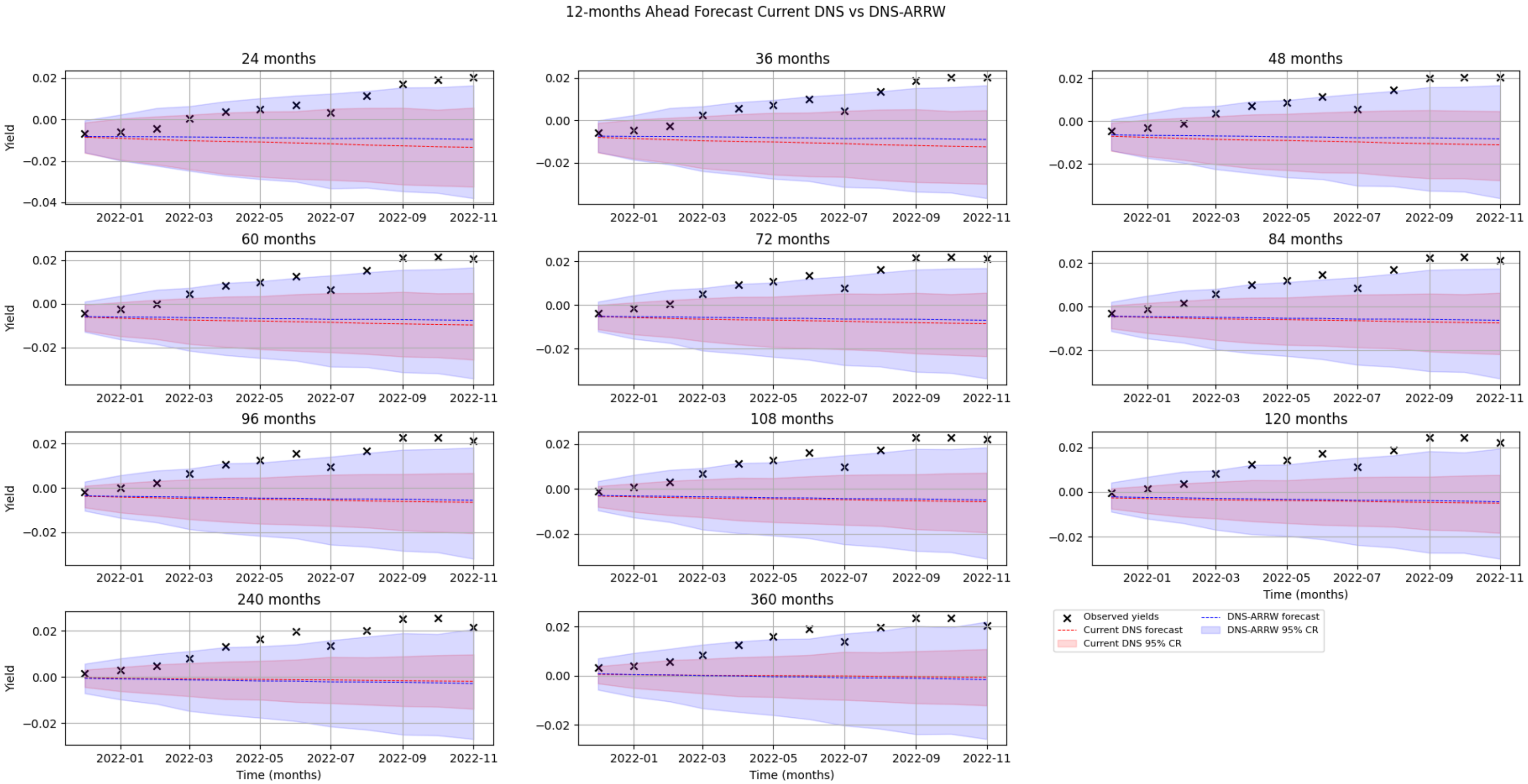


Figure 7.67: The 12-months ahead forecasts of the current model (Current DNS) and the DNS-ARRW model with their respective 95% credible region (CR) based on 1000 path simulations.

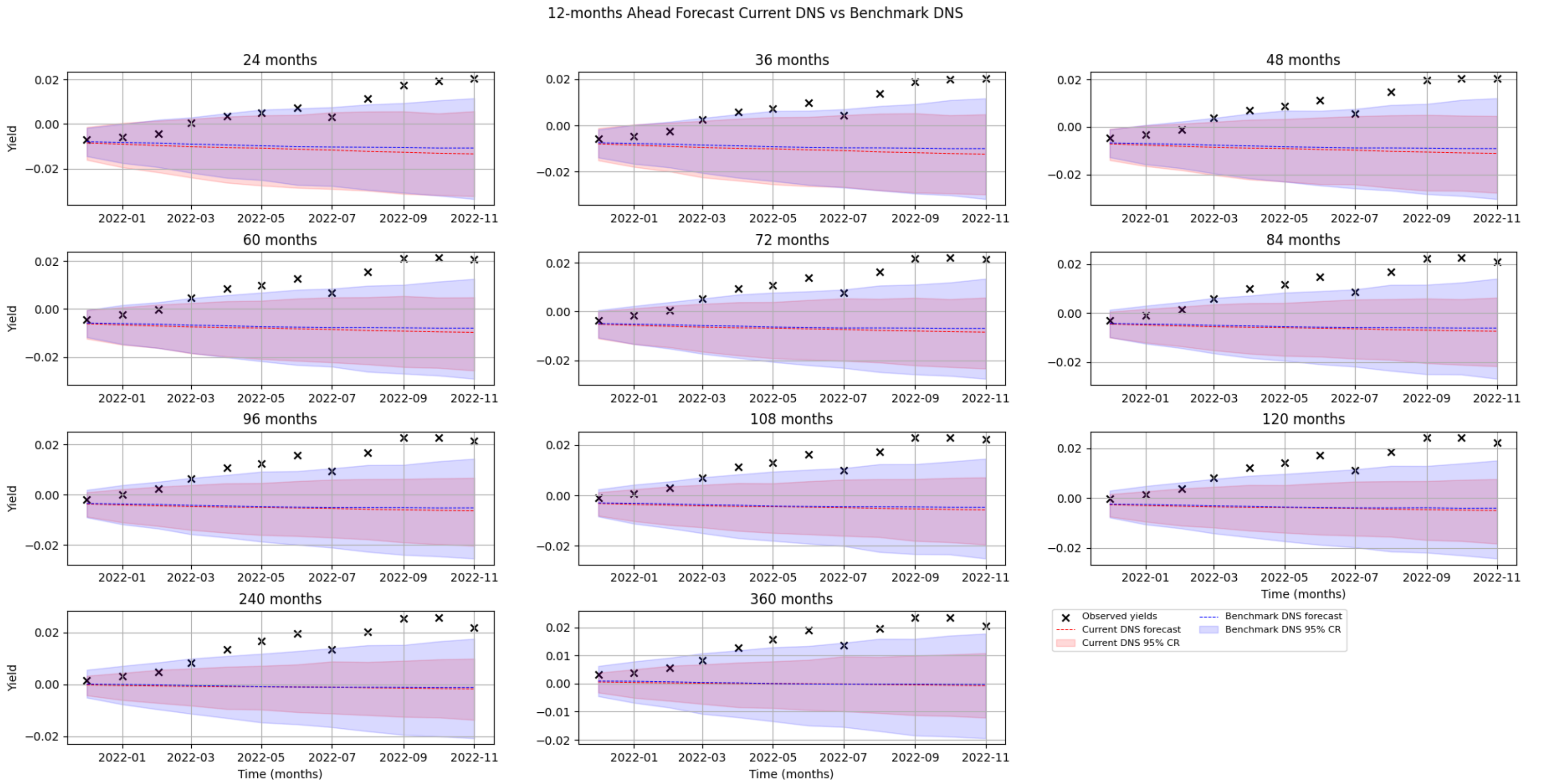


Figure 7.68: The 12-months ahead forecasts of the current model (Current DNS) and the Benchmark DNS model with their respective 95% credible region (CR) based on 1000 path simulations.

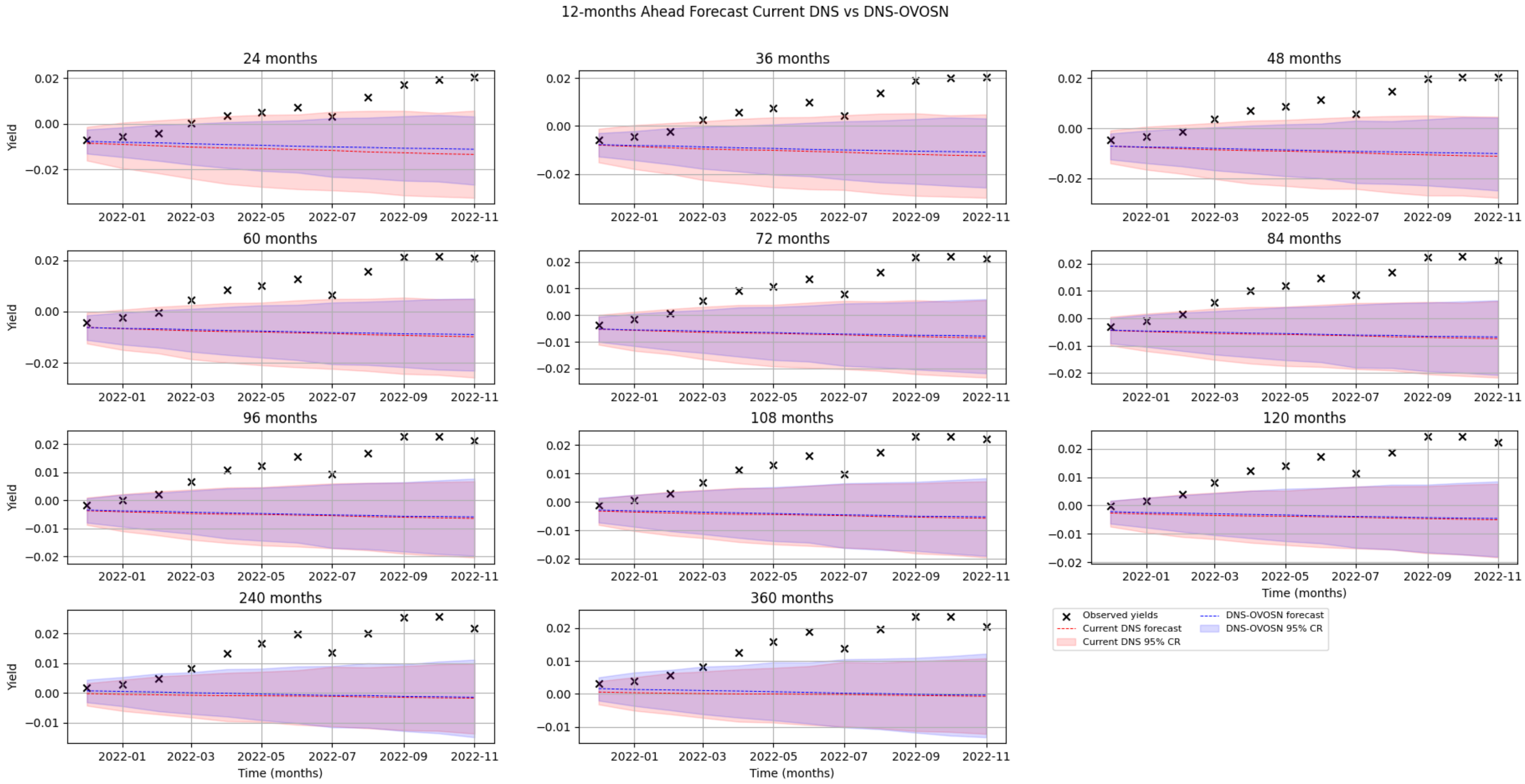


Figure 7.69: The 12-months ahead forecasts of the current model (Current DNS) and the DNS-OVOSN model with their respective 95% credible region (CR) based on 1000 path simulations.

8

Conclusion and Discussion

In this final chapter we provide the answer to our main research question in the conclusion in Section 8.1. Then, in Section 8.2 we discuss some of the limitations of our research and some recommendations on further research that stem from these limitations.

8.1 Conclusion

The aim of this thesis was to research whether extending the current DNS model employed by the DSTA with volatility modeling and using a Bayesian approach to modeling yields can improve yield forecasts. In order to model volatility in a DNS type of model, we started with a baseline state-space model, the benchmark DNS model that resembles the current model the closest. Then, we used the findings of the in-sample and forecasting results to explore modifications to the benchmark DNS model and we used the findings from the literature review on modeling volatility in the DNS framework to arrive at used the volatility models. Moreover, the Bayesian approach on modeling yield curves allowed us to incorporate additional uncertainty from the parameters. Subsequently, we will proceed with answering the subquestions and the main research question.

Subquestion 1: How to model bond yields and how can we extend those models with stochastic volatility?

When incorporating stochastic volatility we had the choice to model volatility as a GARCH process or as a SV process. In addition, we also had the choice to model the volatility through either the observation noise or through the state noise. In the end, we chose to model the volatility through the observation noise as a GARCH process based on a paper of Koopman et al. (2010) that assumes one common volatility process. In this type of model, every maturity has its own volatility loading translating the amount of common volatility that a specific maturity is subject to. This model allowed us to use the Kalman filter in a slightly modified way, resulting into quite efficient state estimation as opposed to more computationally demanding methods as Sequential Monte Carlo or a particle filter.

Subquestion 2: How can a Bayesian approach improve uncertainty quantification in yield curve forecasts?

The Bayesian approach to yield curve modeling means that we assumed explicit uncertainty of the model parameters. Then, using the Random Walk Metropolis algorithm we obtained samples from the approximated posterior distribution, which allowed us to simulate multiple-steps ahead forecasts. We have seen that the additional parameter uncertainty resulted into more variability of the yields, especially resulting into credible regions that had more weight in the increasing yields. Moreover, for more complex models we have seen that the RWM algorithm was able to find parameter sets that had better in-sample performance than the MLE-based estimation of the parameters.

Subquestion 3: How do the different models and methods compare?

Overall, the current model and method (current DNS) performed relatively better than expected compared with the explored models. In particular, the current DNS had an in-sample performance that was mid-range, but still outperforming the basic models like the benchmark DNS and DNS-SN model. For the longer-term maturities the current DNS model even had better in-sample performance than the last volatility model DNS-OVOSN, with a similar overall performance. In contrast, the DNS-ARRW and DNS-OV model outperformed the current DNS model significantly in-sample. Subsequently, the advantage of the Bayesian forecasting method became more clear when we considered the 12-months ahead forecasts. The benchmark DNS, DNS-ARRW and DNS-OV models were able to better capture the uncertainty of especially the increasing yields in their worst-case forecasts. Moreover, the DNS-SN and DNS-OVOSN models did not have better worst-case forecasts, but it seems that these models at least simulated less steeply decreasing yields compared with the current DNS.

Main Research Question: How to model interest rate volatility and quantify forecasting uncertainty with a Bayesian approach in order to forecast interest rate costs better?

First, considering the difference between methods, we conclude that it seems that the Bayesian approach to parameter estimation and forecasting is able to account for more variability in yields and also seems to simulate the direction of the forecasts slightly better than the current MLE-based method for most models. Then, considering the models with a GARCH volatility process, the results are mixed. Although the volatility models DNS-OV and DNS-OVOSN seem to have a good in-sample performance, the forecasting power of the DNS-OVOSN is similar to the current model and the DNS-OV model is still outperformed by the benchmark DNS and the DNS-ARRW models. Of course, we did not consider *all* parameters of the volatility models to have uncertainty, since we fixed the volatility loadings. Moreover, grouping the volatility loadings with other maturity combinations than for the DNS-OVOSN model might also lead to better forecasts, but we will elaborate more on these two points in the discussion. So, we argue that extending the DNS model with volatility modeling has potential for better forecasts compared with the current DNS model with the aforementioned caveats. All in all, the DNS-ARRW model has been the only model that consistently has good in-sample and forecasting performance and significantly improved the 12-months ahead forecasts.

8.2 Discussion and Further Research

This brings us to the last section of this thesis. Throughout our research we have had to make decisions and assumptions in the explored models and the used MCMC and state estimation algorithms. In this section we discuss the limitations of those decisions and we recommend further research that could improve the Bayesian yield curve modeling and forecasting methods that we have used.

First, we fixed the volatility loadings Γ for both volatility models at their MLE values because of difficult convergence of the corresponding chains when using the Random Walk Metropolis algorithm. As a result, eleven parameters of the DNS-OV and three parameters of the DNS-OVOSN model did not contribute to the parameter uncertainty in the one-month and 12-months ahead forecasts. We have seen for the DNS-ARRW model that the additional parameter uncertainty can result into more variability in yield simulations. In addition, the volatility models seem to capture the direction of the forecasts slightly better than the other models. So, considering the parameter uncertainty of the volatility loadings seems promising. Then, there are two ways the parameter identification could be tackled in a better way. First, we could consider fixing only one of the volatility loadings as they are proportions relative to each other and include the remainder of the volatility loadings in the RWM run in order to have some baseline proportion value. Additionally, we could use more advanced MCMC algorithms. In particular, the so-called *Metropolis-Adjusted Langevin Algorithm*, or MALA in short, could be considered. This MCMC algorithm involves using gradients at each iteration and is in general better-suited for high-dimensional parameter spaces than the RWM algorithm.

In addition, we have used the total RMSE of the worst-case forecasts of every model. Since these are the upper bounds of the 95% credible regions of the forecasting simulations, the RMSE does not take into account whether the observed yields are “inside” or “outside” the credible regions. Arguably, the error stemming from observed yields inside the credible regions are preferable over observations being outside the credible regions as this means that the observation is included in the forecast uncertainty. Subsequently, the lower bound of the credible regions or the direction of the forecast is not directly accounted for in the current forecast performance measure. So, it could be better to use tools specifically developed for forecasts comparisons. One such tool is the Diebold-Maiorano test, which tests whether two forecasts differ significantly.

Then, we have chosen to extend the benchmark DNS model with a GARCH volatility process through the observation noise. The choice for a GARCH process proposed by Koopman et al. (2010) was mainly driven by the fact that we could use the Kalman filter without too many modifications, which is relatively computationally fast and does not entail too many estimations compared with Sequential Monte Carlo or particle filters. Additionally, modeling the volatility through the observation noise was driven by the promising results of the original authors compared with the state noise variant. However, a more promising and more intuitive extension would be a SV(1) volatility process through the state noise from a modeling perspective as we noted in the concluding remarks of the literature review. Such a model allows for *actual* stochastic volatility that is its own stochastic process instead of conditionally deterministic. However, this comes at cost of more approximations as the likelihood cannot typically be derived analytically for SV(1) models.

Moreover, we have assumed that a better model includes volatility since we have seen that bond yields have been quite volatile in the past one to two years. So, our assumption is mostly driven by market observations. However, modeling volatility does not consider the underlying cause of the volatility, which are the interest rate hikes of the European Central Bank. Specifically, the interest rate hikes or decreases are fundamental for bond yield increases or decreases. So, especially for monthly data it could be reasonable that volatility has less effect on the underlying dynamics of bond yields compared with actual interest rate “jumps”. Consequently, two directions could be worth exploring. First, it could be interesting to incorporate the stochastic volatility model with jumps (Hautsch and Ou, 2008a), or SVJ in short, which extends the SV(1) process with a Normally distributed *jump magnitude* variable multiplied with a Bernoulli *jump occurrence* variable. Another way of incorporating the interest rate hikes, or any macro-economical process, could be by augmenting it to the state variable or adding the effect directly as some external variable to the state-space model.

Subsequently, we have assumed Gaussian noise terms for all DNS type of models. However, it is readily known that for many financial processes extreme events are more likely to happen than the Normal distribution accounts for. So, perhaps non-Gaussian noise terms can also increase forecasting power without having to resort to more complex volatility or jump processes. For non-Gaussian models one could think of applying Extreme Value Theory by using the Generalized Extreme Value (GEV) distribution, or in general of the Student’s-t distribution. Particle filters proposed by Kitagawa (1996) could be considered for such state-space models.

Furthermore, we have only considered the DNS type of models for modeling yield curves from the beginning. Naturally, other methods exist as well to model yield curves. A lot of literature exists on interest rate models, particularly in the context of pricing models. The reason for staying in the DNS framework is driven by the fact that it is “parsimonious” as the original authors (C. R. Nelson and Siegel, 1987) call their original model, so it contains relatively few variables which makes these type of models easy to interpret. However, other methods besides the DNS type of models could have better forecasting performance.

Lastly, as mentioned in Section 2.4 we have used the direct bond yield data without discounting the cash flow since we are mostly interested in bond yield trends observed in the market. So, the results of this thesis cannot be used for direct pricing of bonds as they do not reflect the time value of money like a zero-coupon yield curve would. So, in further research it would be preferable to prepare the yield data well, especially if the goal is to use the modeled yields for portfolio strategies or pricing interest rate related instruments.

Bibliography

Anderson, B. D. O., & Moore, J. B. (1979). *Optimal filtering* (T. Kailath, Ed.). Prentice-Hall.

Bailer-Jones, C. A. L. (2017, July). *Parameter estimation: Single parameter*. Cambridge University Press. https://doi.org/10.1017/9781108123891

Bar-Shalom, Y., Li, X.-R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. John Wiley & Sons. https://doi.org/10.1002/0471221279

Berk, J. B., DeMarzo, P. M., & Harford, J. V. T. (2021). *Fundamentals of corporate finance* (5th ed.). Pearson Education.

Bishop, C. M. (2006). *Pattern recognition and machine learning* (M. Jordan, J. Kleinberg, & B. Schölkopf, Eds.). Springer.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*, 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Brockwell, P. J., & Davis, R. A. (1991). *Time series: Theory and methods* (2nd ed.). Springer.

Christensen, J. H., Diebold, F. X., & Rudebusch, G. D. (2011). The affine arbitrage-free class of nelson-siegel term structure models. *Journal of Econometrics*, *164*, 4–20. https://doi.org/10.1016/j.jeconom.2011.02.011

De Vivo, F., Brandl, A., Battipede, M., & Gili, P. (2017). Joseph covariance formula adaptation to square-root sigma-point kalman filters. *Nonlinear Dynamics*, *88*, 1969–1986. https://doi.org/10.1007/s11071-017-3356-x

Diebold, F. X., & Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics*, *130*, 337–364. https://doi.org/10.1016/j.jeconom.2005.03.005

Diebold, F. X., Rudebusch, G. D., & Aruoba, S. B. (2006). The macroeconomy and the yield curve: A dynamic latent factor approach. *Journal of Econometrics*, *131*, 309–338. https://doi.org/10.1016/j.jeconom.2005.01.011

DSTA. (2023). *Outlook 2024* [Accessed: December 19, 2023]. https://www.dsta.nl/documenten/publicaties/2023/12/15/outlook-2024

Duffee, G. R. (2012). Forecasting interest rates. In *Handbook of economic forecasting* (Vol. 2). Elsevier.

Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods* (2nd ed.). Oxford University Press. http://www.oup.co.uk/academic/science/maths/series/osss/

ECB. (2023). *Key ecb interest rates* [Accessed: December 28, 2023]. https://www.ecb.europa.eu/stats/policy_and_exchange_rates/key_ecb_interest_rates/html/index.en.html

Fama, E. B., & Bliss, R. R. (1987). The information in long-maturity forward rates. *The American Economic Review*, *7*, 680–692.

Filipovic, D. (2009). *Term-structure models* (1st ed.). Springer Berlin. https://doi.org/https://doi.org/10.1007/978-3-540-68015-4

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (3rd ed.). Taylor & Francis.

Gerlach, R., & Tuyl, F. (2006). Mcmc methods for comparing stochastic volatility and garch models. *International Journal of Forecasting*, *22*, 91–107. https://doi.org/10.1016/j.ijforecast.2005.04.020

Geweke, J. (1991, December). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Federal Reserve Bank of Minneapolis. https://www.researchgate.net/publication/2352607

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance*, *48*, 1779–1801. https://doi.org/10.1111/j.1540-6261.1993.tb05128.x

Harvey, A. C. (1990, July). State space models and the kalman filter. In *Forecasting, structural time series models and the kalman filter* (pp. 100–167). Cambridge University Press. https://doi.org/10.1017/cbo9781107049994.004

Harvey, A. C., Ruiz, E., & Sentana, E. (1992). Unobserved component time series models with arch disturbances. *Journal of Econometrics*, *52*, 129–157. https://doi.org/10.1016/0304-4076(92)90068-3

Hautsch, N., & Ou, Y. (2008a). *Discrete-time stochastic volatility models and mcmc-based statistical inference **. http://ssrn.com/abstract=1292494

Hautsch, N., & Ou, Y. (2008b). *Yield curve factors, term structure volatility, and bond risk premia yield curve factors, term structure volatility, and bond*. http://sfb649.wiwi.hu-berlin.de

Hautsch, N., & Yang, F. (2012). Bayesian inference in a stochastic volatility nelson-siegel model. *Computational Statistics and Data Analysis*, *56*, 3774–3792. https://doi.org/10.1016/j.csda.2010.07.003

Ibáñez, F. (2015). *Calibrating the dynamic nelson-siegel model: A practitioner approach*. Bank of Chile.

Kalman, R. E. (1960). On the general theory of control systems. *IFAC Proceedings Volumes*, *1*, 491–502. https://doi.org/10.1016/S1474-6670(17)70094-8

Kim, C.-J., & Nelson, C. R. (1999). *State-space models with regime switching: Classical and gibbs-sampling approaches with applications* (1st ed.). MIT Press. https://doi.org/https://doi.org/10.7551/mitpress/6444.001.0001

Kim, S., & Shephard, N. (1998). Stochastic volatility : Likelihood inference and comparison with arch models. *Review of Economic Studies*, *65*, 361–393. https://doi.org/https://doi.org/10.1111/1467-937X.00050

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*, 1–25. https://doi.org/10.2307/1390750

Koopman, S. J., Mallee, M. I. P., & Van der Wel, M. (2010). Analyzing the term structure of interest rates using the dynamic nelson-siegel model with time-varying parameters. *Journal of Business and Economic Statistics*, 329–343.

Kreuzer, A., & Czado, C. (2020). Efficient bayesian inference for nonlinear state space models with univariate autoregressive state equation. *Journal of Computational and Graphical Statistics*, *29*(3), 523–534. https://doi.org/10.1080/10618600.2020.1725523

Kumar, A., Mallick, S., Mohanty, M. S., & Zampolli, F. (2022). *Bis working papers no 606: Market volatility, monetary policy and the term premium*. Bank for International Settlements. https://www.bis.org/publ/work606.pdf

Lee, C.-F., & Lee, J. C. (Eds.). (2015). *Handbook of financial econometrics and statistics*. Springer.

Martínez-Hernández, I., Gonzalo, J., & González-Farías, G. (2022). Nonparametric estimation of functional dynamic factor model. *Journal of Nonparametric Statistics*, *34*, 895–916. https://doi.org/10.1080/10485252.2022.2080825

Mesters, G., Schwaab, B., & Koopman, S. J. (2014). A dynamic yield curve model with stochastic volatility and non-gaussian interactions: An empirical study of non-standard monetary policy in the euro area. http://www.tinbergen.nl

Nelson, C. R., & Siegel, A. F. (1987). Parsimonious modeling of yield curves. *The Journal of Business*, *60*, 473–489. https://www.jstor.org/stable/2352957

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Source: Econometrica*, *59*, 347–370. https://doi.org/10.2307/2938260

Oosterlee, C. W., & Grzelak, L. A. (2019, November). *Mathematical modeling and computation in finance: With exercises and python and matlab computer codes*. World Scientific Europe.

Preminger, A., & Hafner, C. M. (2010). Deciding between garch and stochastic volatility via strong decision rules. *Journal of Statistical Planning and Inference*, *140*, 791–805.

Rijksoverheid. (2023). *Miljoenennota 2024* [Accessed: December 19, 2023]. https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/begrotingen/2023/09/19/miljoenennota-2024/Miljoenennota-2024.pdf

Robert, C. P., & Casella, G. (2004). *Monte carlo statistical methods* (2nd ed.). Springer.

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, *16*, 351–367.

Shumway, R. H., & Stoffer, D. S. (2011). *Time series analysis and its applications* (G. Casella, S. Fienberg, & I. Olkin, Eds.; 3rd ed.). Springer. www.springer.com/series/417

Sontag, E. D. (1998). *Mathematical control theory* (2nd ed.). Springer.

Svensson, L. E. O. (1994). *Estimating and interpreting forward interest rates: Sweden 1992-1994*. National Bureau of Economical Research.

Taleb, N. N. (2007). *Fooled by randomness: The hidden role of chance in life and in the markets* (2nd ed.). Penguin Books.

Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes - a study of daily sugar prices, 1961-79. In O. D. Anderson (Ed.), *Time series analysis: Theory and practice* (pp. 203–226, Vol. 1). Elsevier/North-Holland.

Van der Meulen, F. (2022). *Statistical inference lecture notes for the course wi4455*. https://github.com/fmeulen/WI4455.

Vivekvinushanth, C. (2020). *Markov and hidden markov model* [Accessed: November 20, 2023]. https://towardsdatascience.com/markov-and-hidden-markov-model-3eec42298d75

Wan, E. A., & Van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 153–158. https://doi.org/10.1109/ASSPCC.2000.882463

Young, G. A., & Smith, R. L. (2005). *Essentials of statistical inference* (1st ed.). Cambridge University Press.

A

Additional Results

In this appendix we provide some additional results for the interested reader. Specifically, in Section A.1 the matrix correlation plots and the running mean of the chains of the RWM runs that result into convergence are shown. In Section A.2 the additional 12-months ahead forecasts are shown of the DNS-SN and the DNS-OV model. Moreover, the relevant subsection, in which these additional results are mentioned, are provided as well.

A.1 RWM Convergence

A.1.1 Benchmark DNS

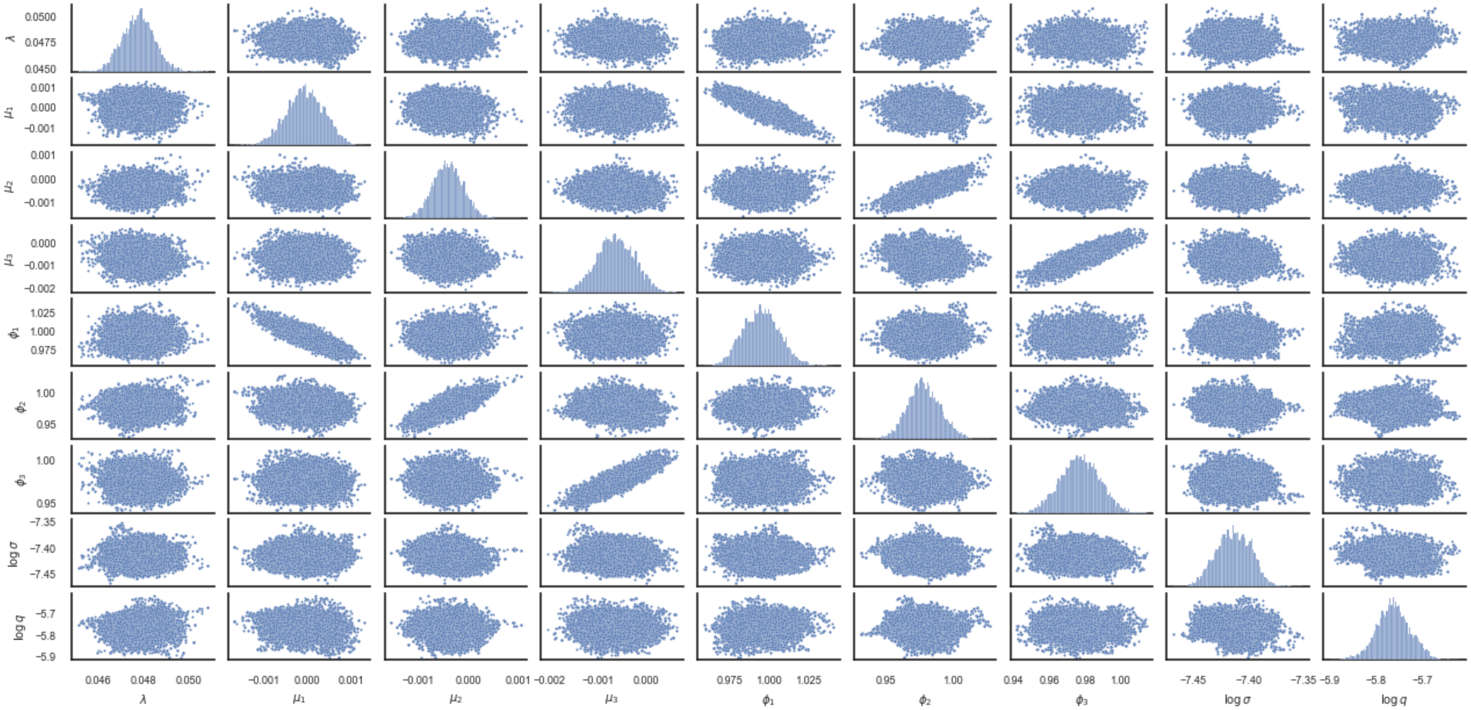


Figure A.1: The matrix correlation plot of the RWM samples of the benchmark DNS model (see Subsection 7.1.1).

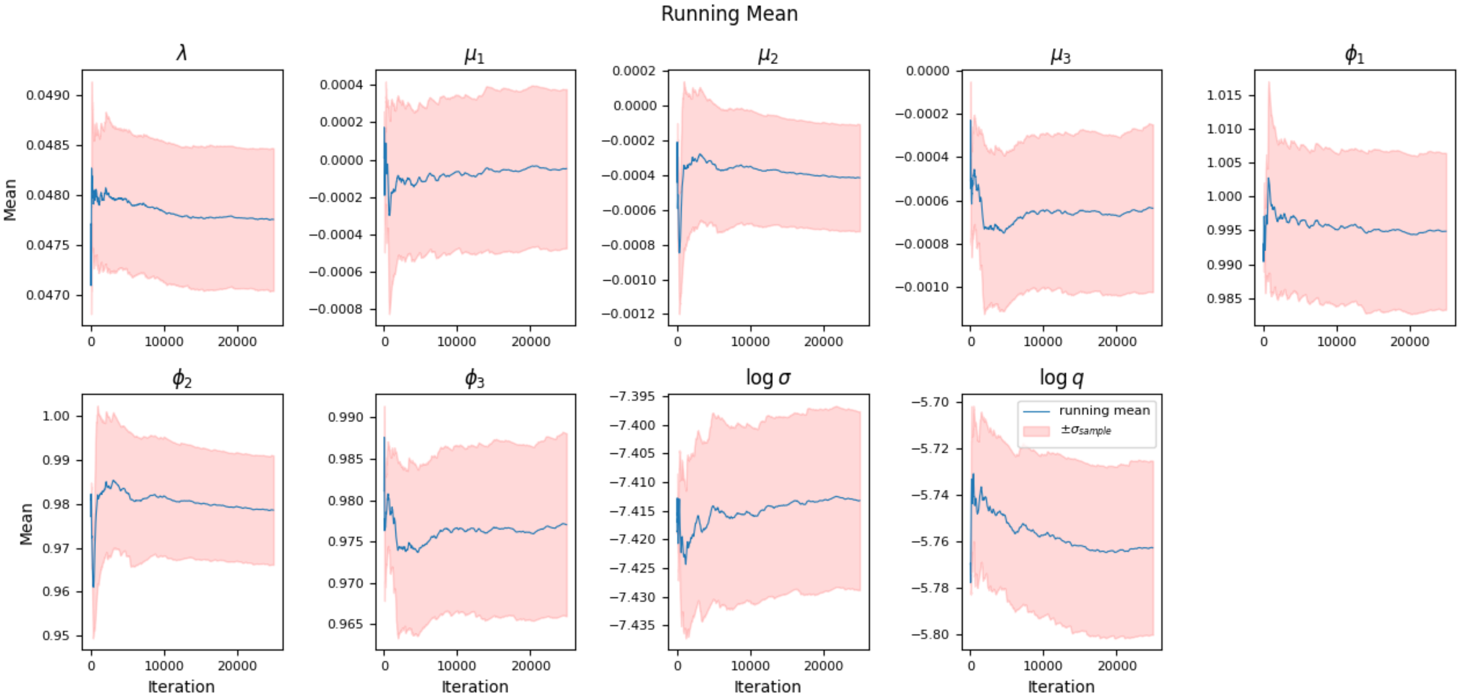


Figure A.2: The running mean of the chains for each parameter of the benchmark DNS model (see Subsection 7.1.1).

A.1.2 DNS-SN

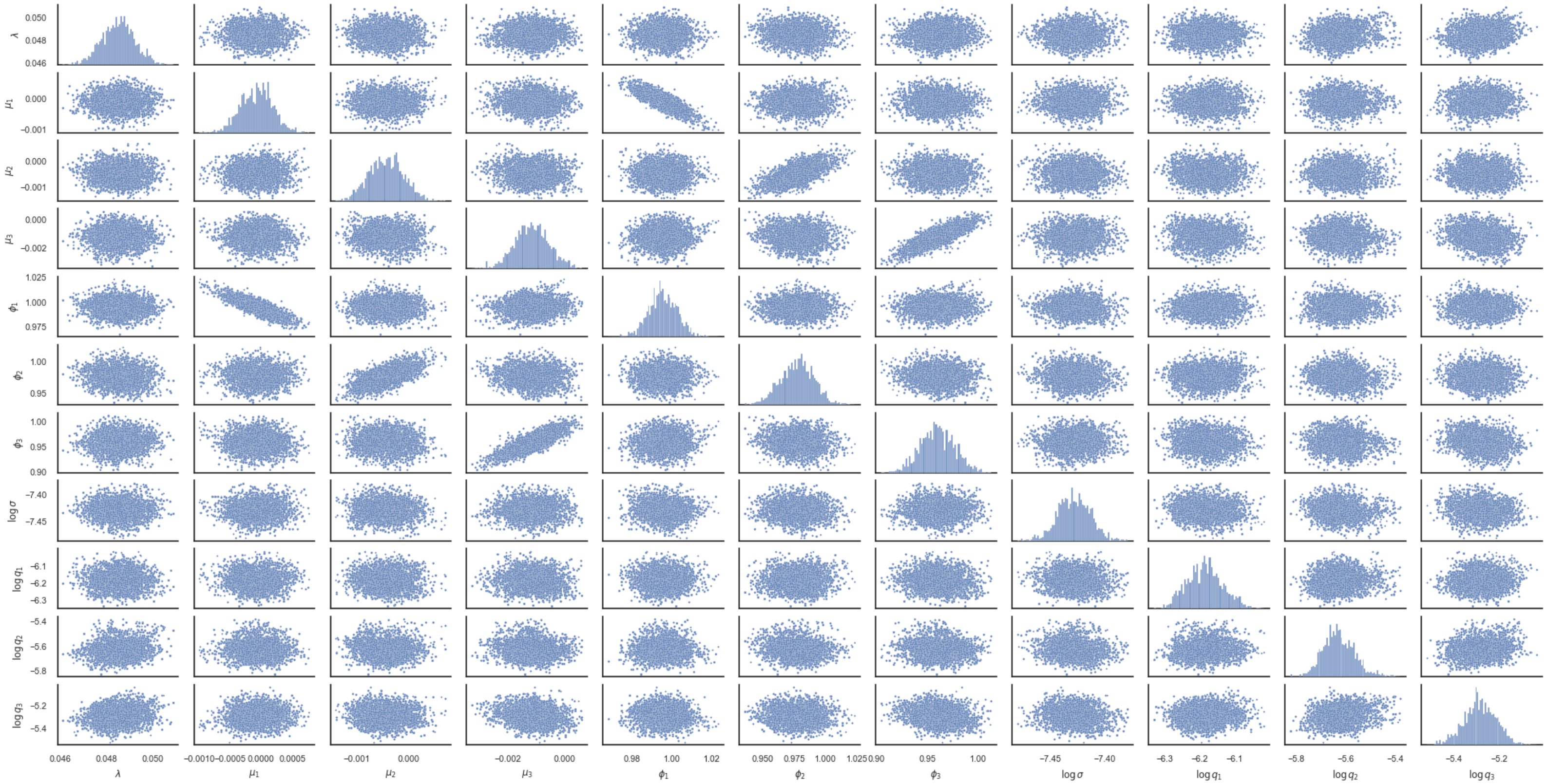


Figure A.3: The matrix correlation plot of the RWM samples of the DNS-SN model (see Subsection 7.2.1).

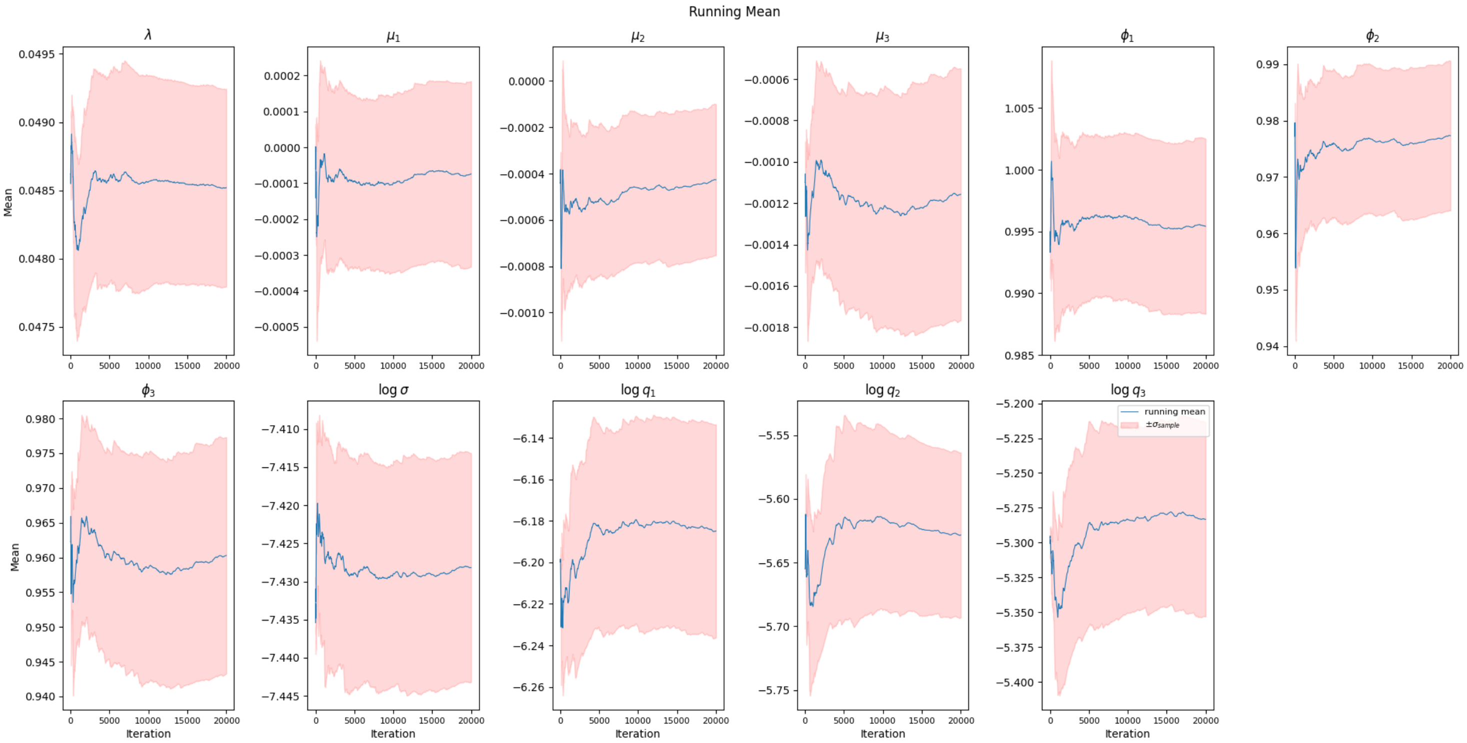


Figure A.4: The running mean of the chains for each parameter of the DNS-SN model (see Subsection 7.2.1).

A.1.3 DNS-ARRW

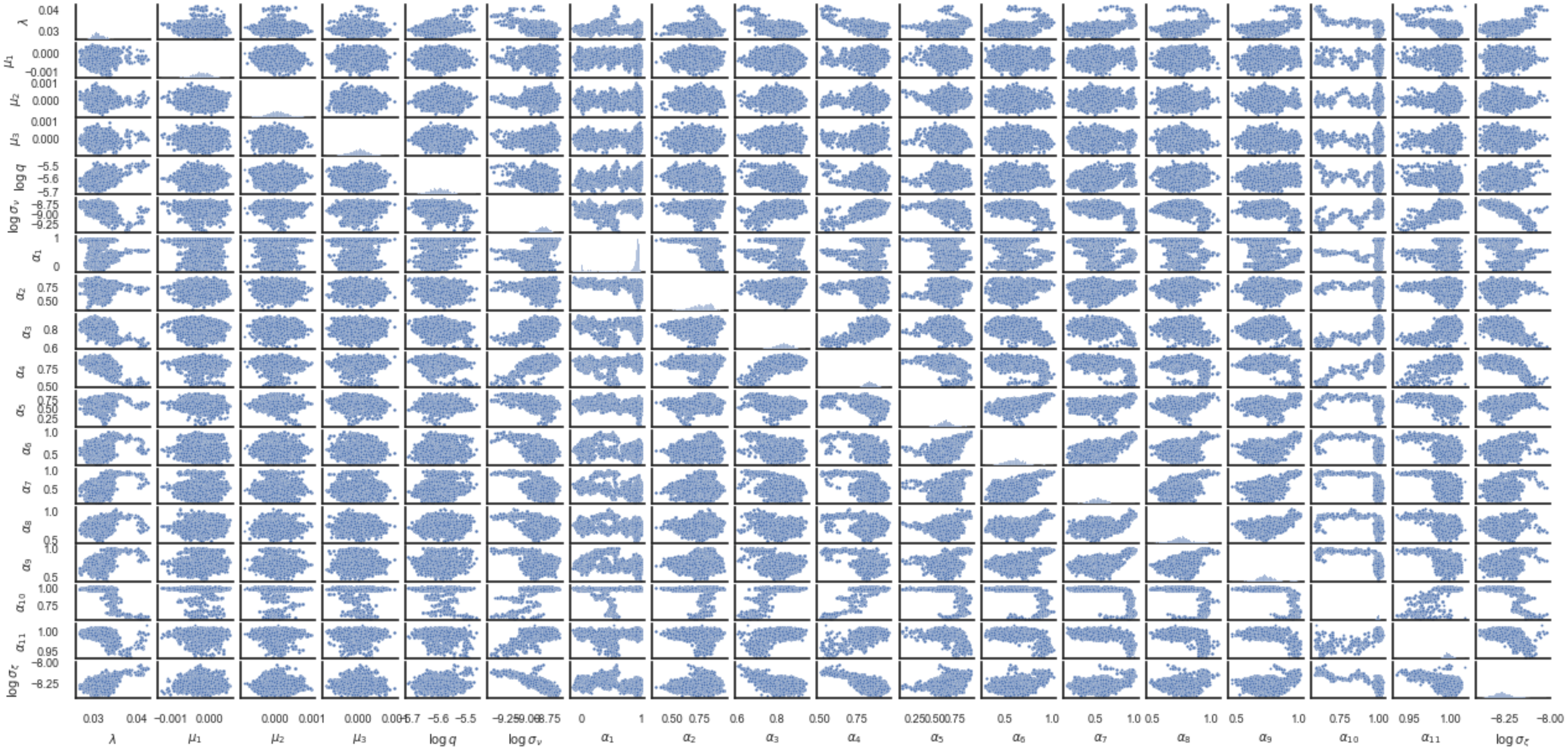


Figure A.5: The matrix correlation plot of the RWM samples of the DNS-ARRW model (see Subsection 7.3.1).

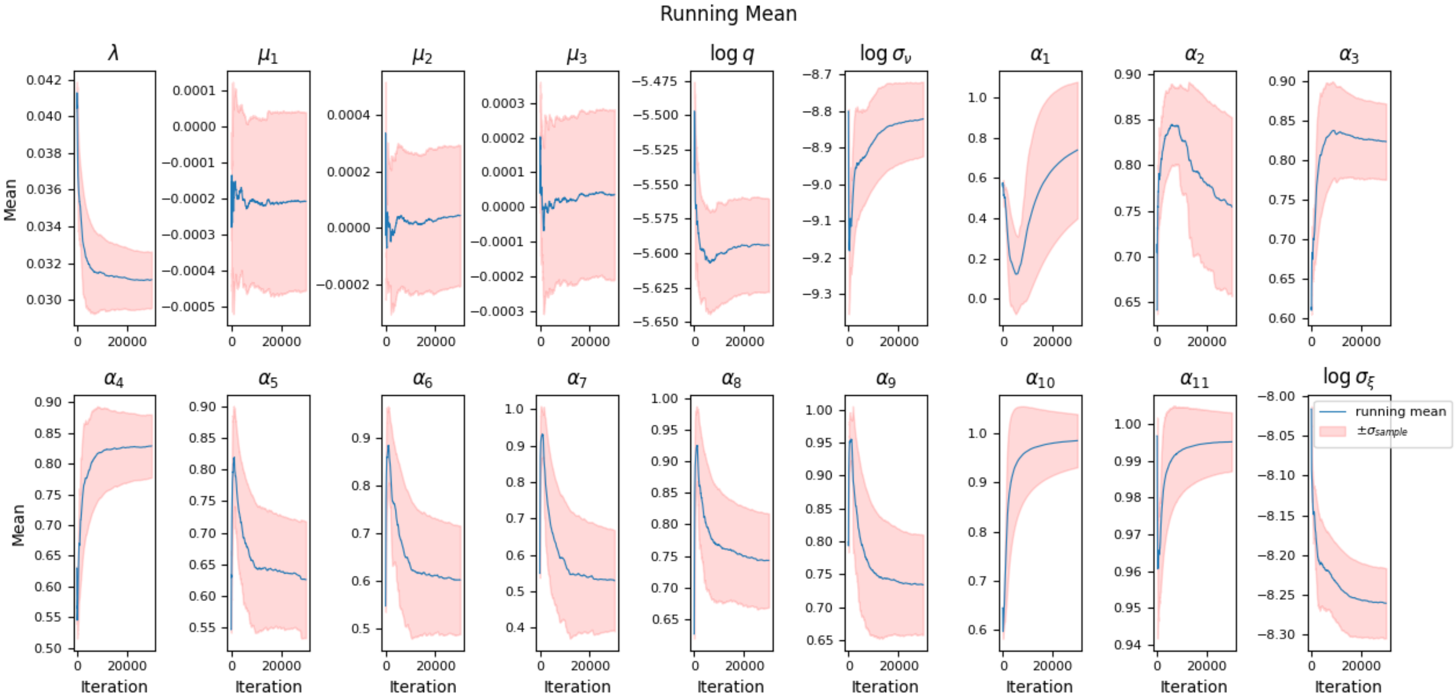


Figure A.6: The running mean of the chains for each parameter of the DNS-ARRW model (see Subsection 7.3.1).

A.1.4 DNS-OV

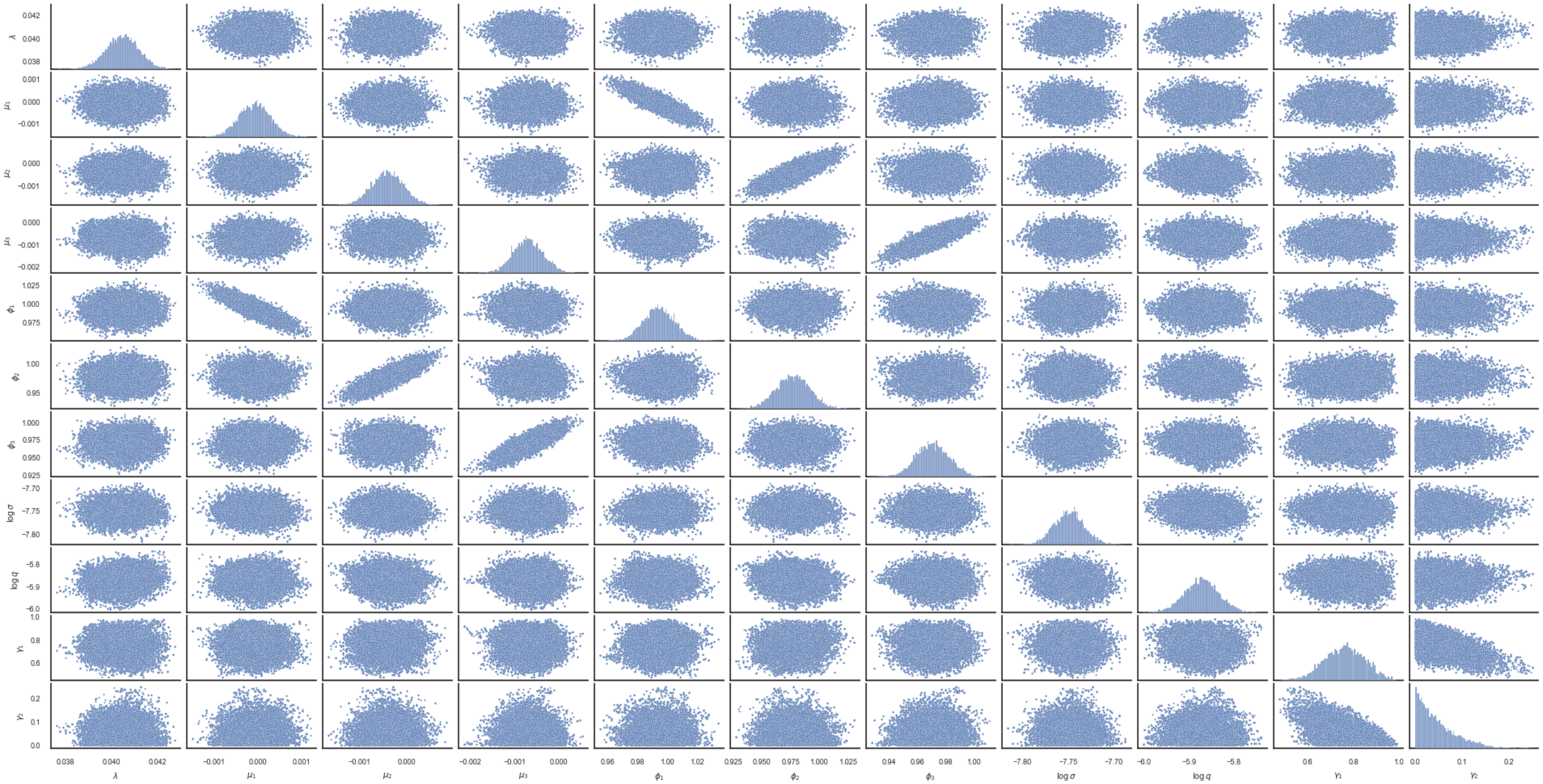


Figure A.7: The matrix correlation plot of the RWM samples of the DNS-OV model (see Subsection 7.4.1).

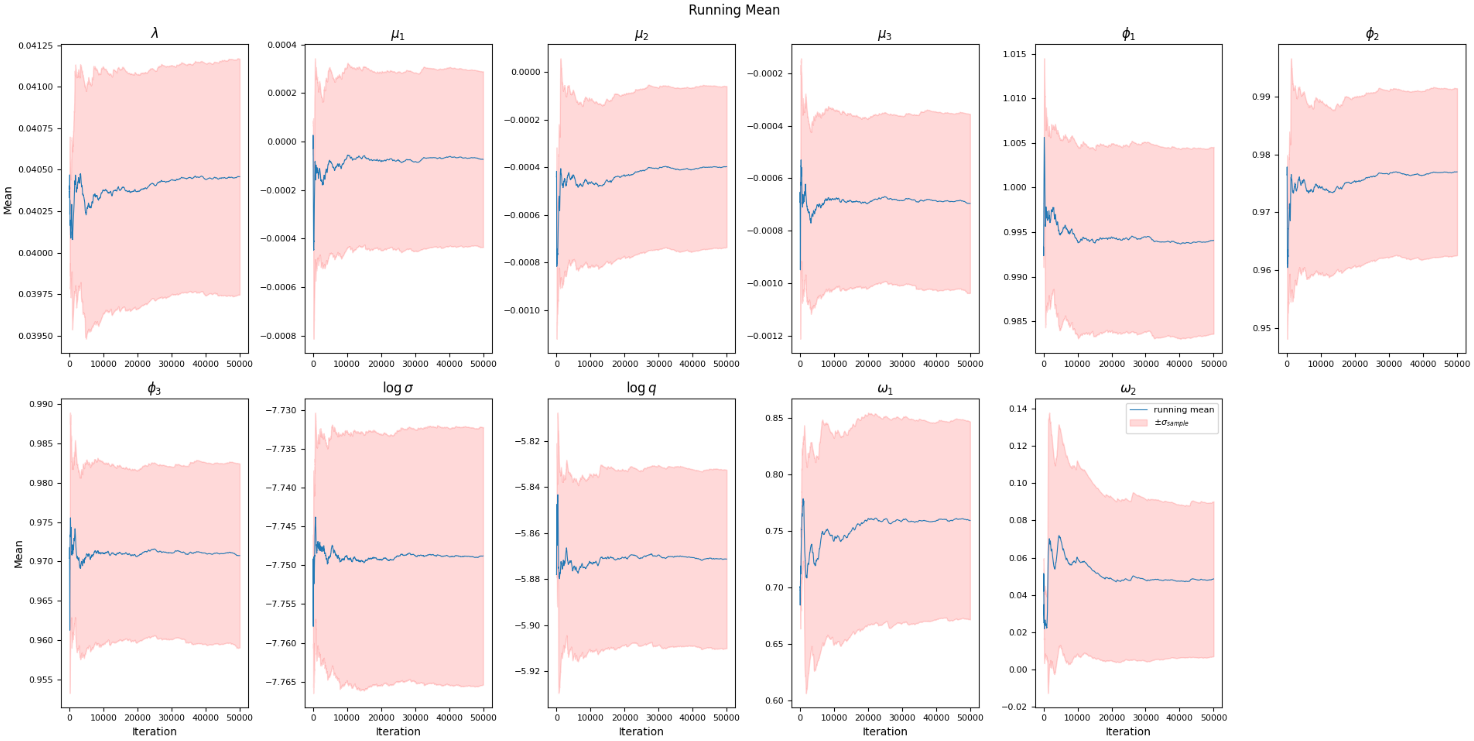


Figure A.8: The running mean of the chains for each parameter of the DNS-OV model (see Subsection 7.4.1).

A.1.5 DNS-OVOSN

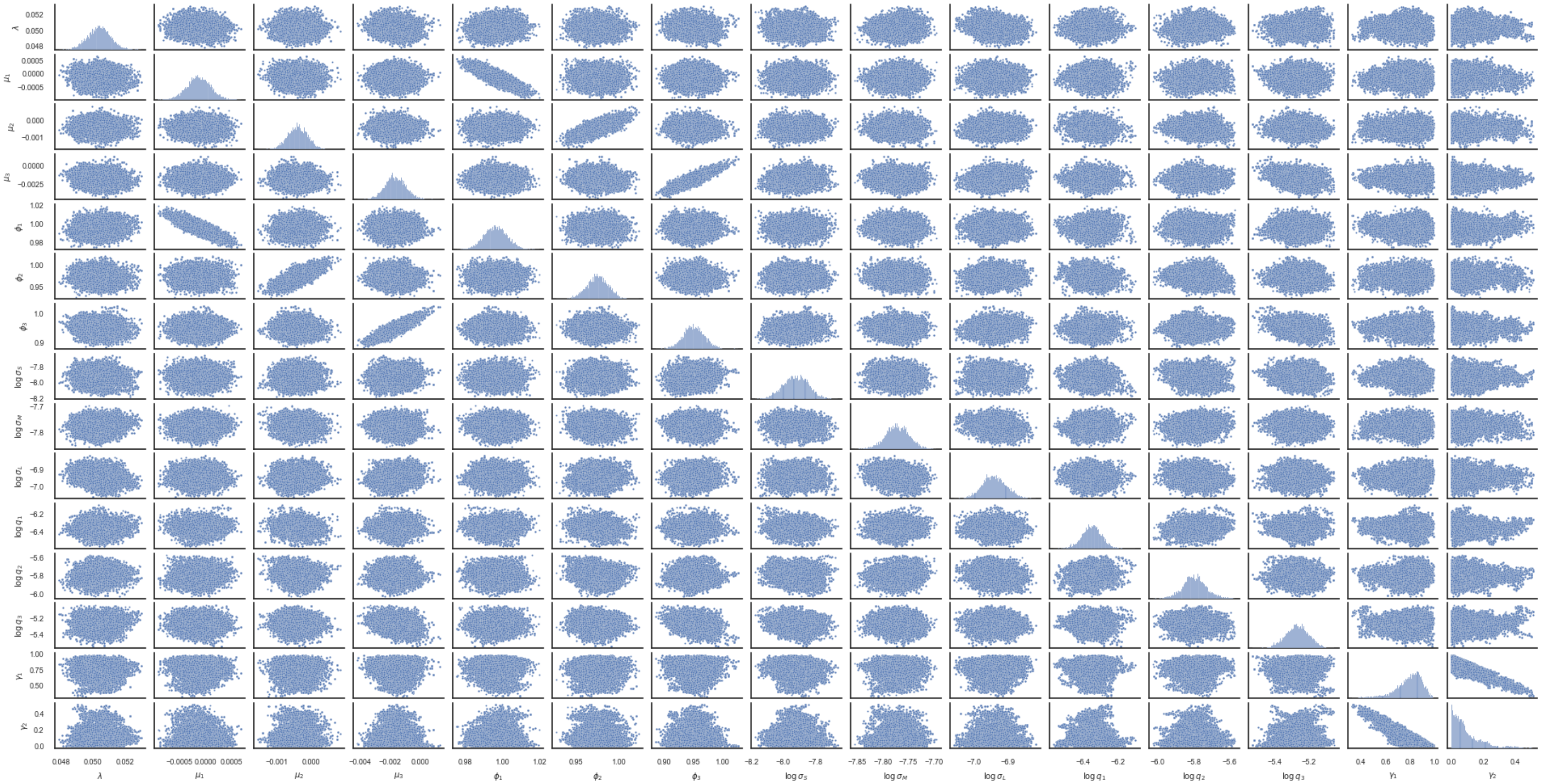


Figure A.9: The matrix correlation plot of the RWM samples of the DNS-OVOSN model (see Subsection 7.5.1).

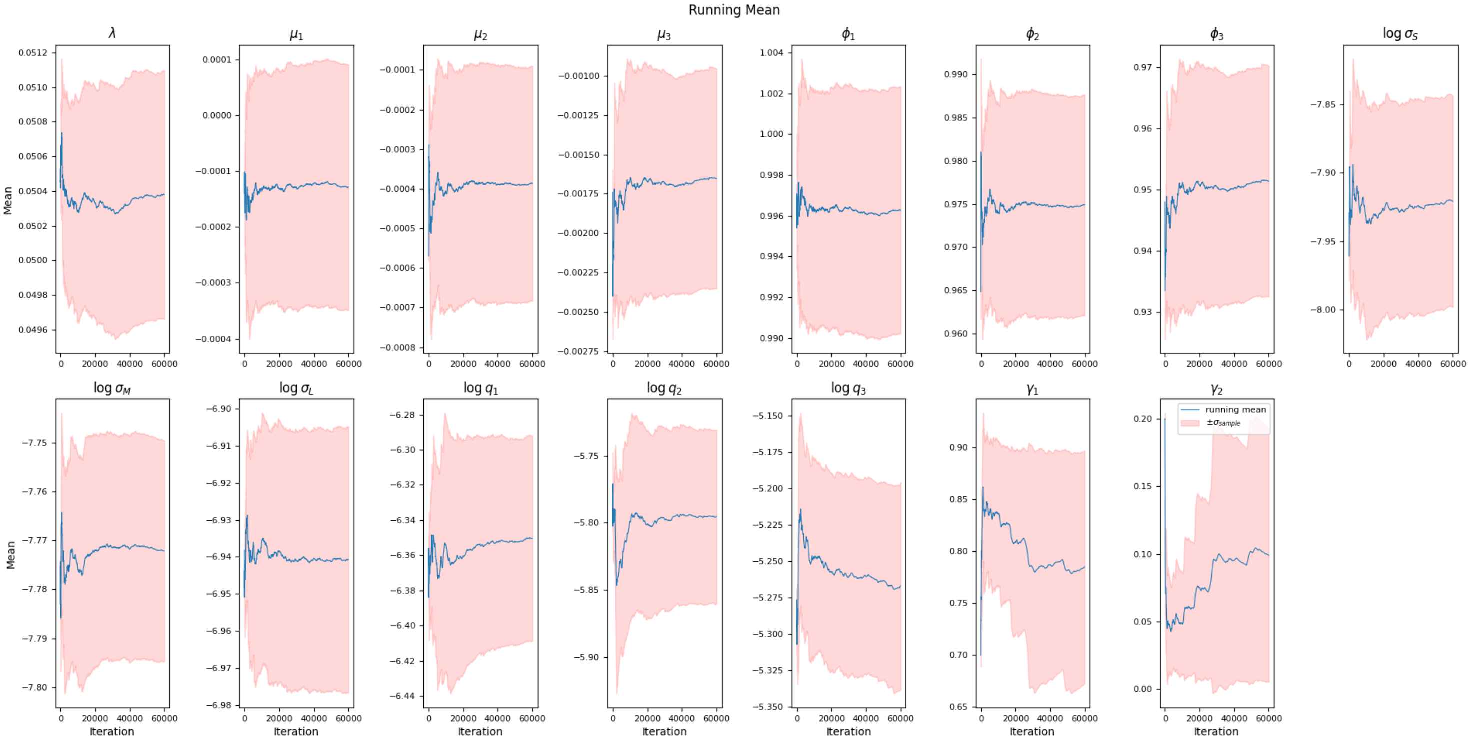


Figure A.10: The running mean of the chains for each parameter of the DNS-OVOSN model (see Subsection 7.5.1).

A.2 Twelve-Months Ahead Forecasts

A.2.1 DNS-SN

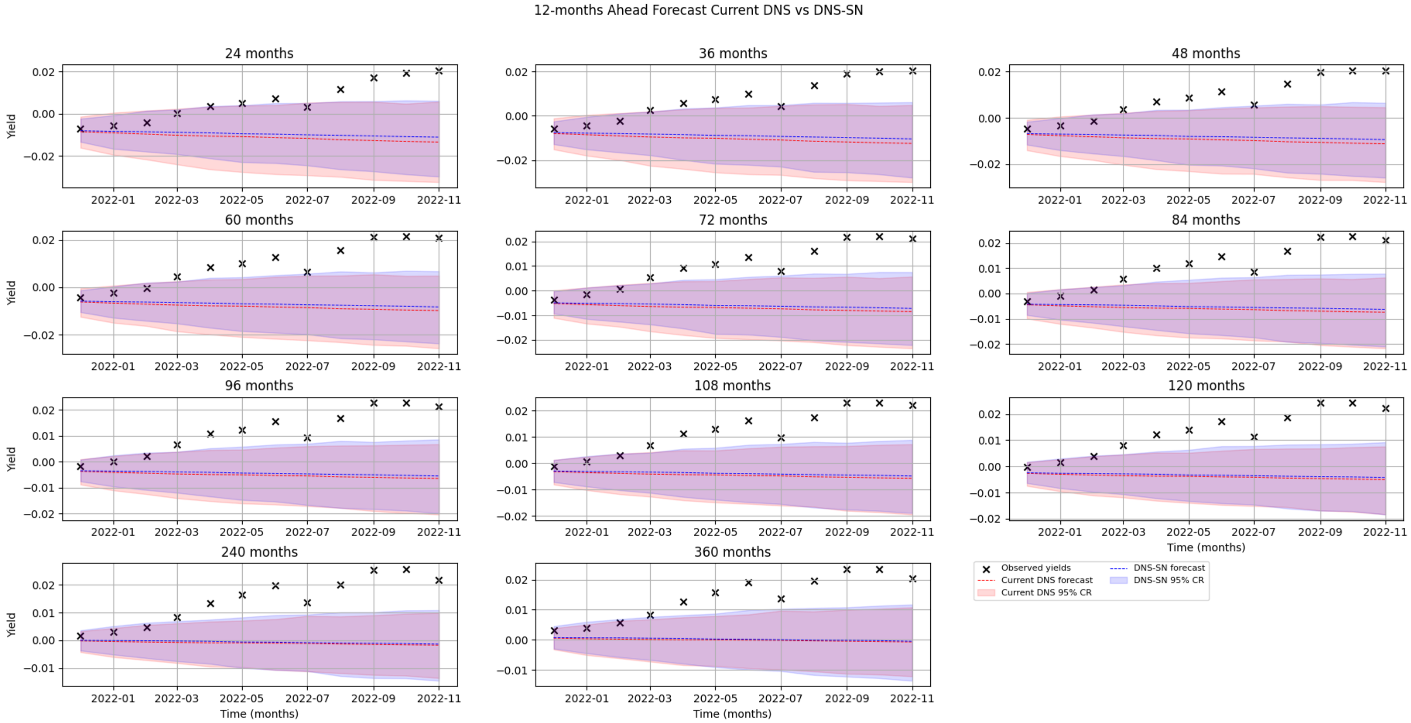


Figure A.11: The 12-months ahead forecasts of the current model (Current DNS) and the DNS-SN model with their respective 95% credible region (CR) based on 1000 path simulations (see Subsection 7.6.2).

A.2.2 DNS-OV

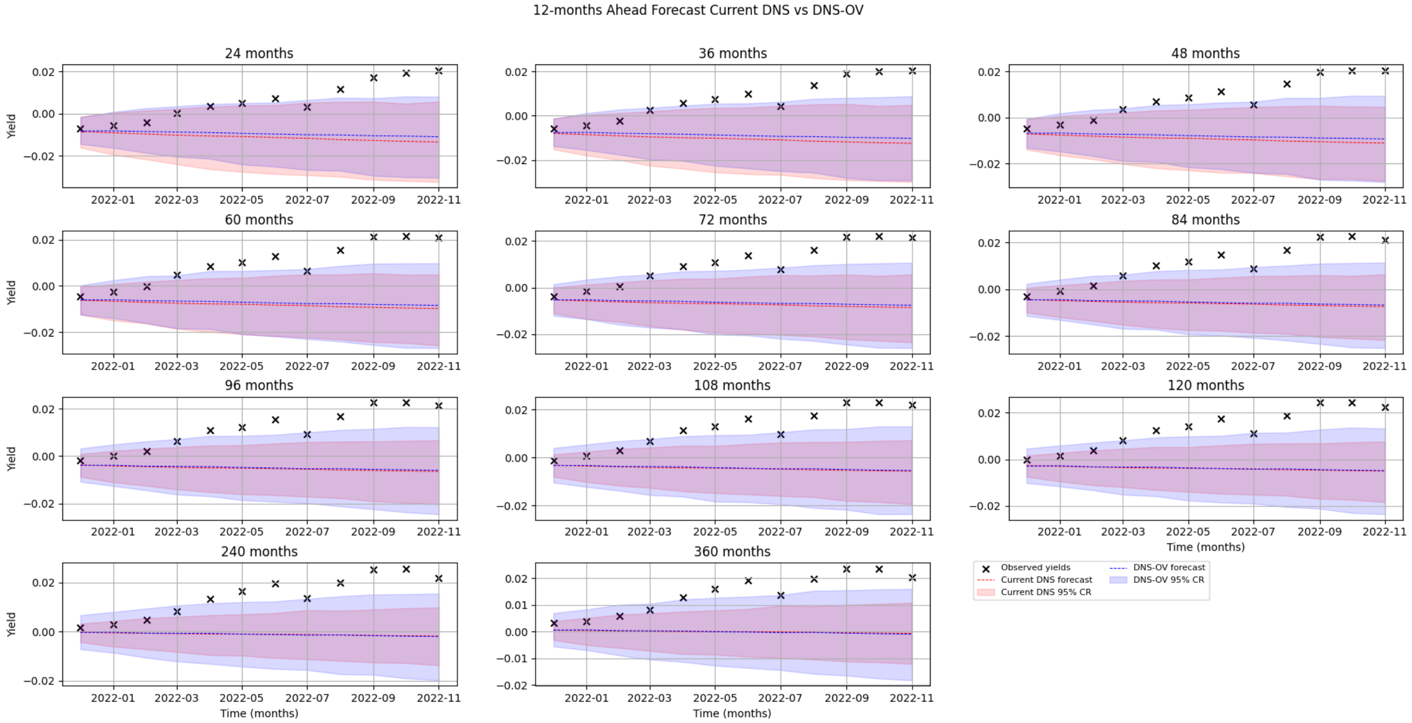


Figure A.12: The 12-months ahead forecasts of the current model (Current DNS) and the DNS-OV model with their respective 95% credible region (CR) based on 1000 path simulations (see Subsection 7.6.2).