TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Macroeconomic factors in interest rate modelling

ausgeführt am

Institut für
Stochastik und Wirtschaftsmathematik
TU Wien

unter der Anleitung von

Associate Prof. Dipl.-Ing. Dr.techn. Stefan Gerhold

durch

Bernhard Knapp, BSc
Matrikelnummer: 1027250
Rennbahnweg 46
1220 Wien

Wien, am 18.06.2020

Abstract

The focus of this thesis is the research on joint-macro models that are appropriate and convenient for the debt strategy and risk management analysis of a Government. The main requirements on such a model are an appropriate modelling and forecasting of the term-structure dynamics as well as diagnostic tools to describe the interactions to relevant macroeconomic factors. Leveraging on a series of working papers published by the Bank of Canada the extended Nelson-Siegel model suggested by Diebold and Li (2006) is introduced as well as various developments on its model specification. In this regards, the Svensson model is introduced as well. The approach used to include macroeconomic factors in a term-structure modelling framework is based on the work from Bolder and Liu (2007) and Diebold et al. (2006).

Finally, in an euro-area environment of interest rates and macroeconomic factors, the models are examined in terms of their ability to capture the dynamics of the term-structure of interest rates, jointly describe the interactions between macroeconomic factors and the term-structure curve, and forecast interest rates.

Contents

1. Introduction

Numerous working papers introduce interest rate term-structure modelling frameworks that incorporate macroeconomic factors. However, the underlying motives of the researchers for modelling the term-structure of interest rates are various as well, inter alia, pricing of financial instruments such as bonds and options, analysing the dynamic interactions among the macro economy and the term-structure of interest rates, or forecast simulation of interest rates for risk management purposes. The main focus of this paper is the research on joint-macro models that are appropriate and convenient for the debt strategy and risk management analysis of a Government. The main requirements on such a model are an appropriate modelling and forecasting of the term-structure dynamics as well as diagnostic tools to describe the interactions to relevant macroeconomic factors. Overall, finance literature indicates valuable benefits of incorporating macroeconomic information in term-structure models, although the majority of the papers have not focused on the practical usage of the models in terms of debt strategy and risk management analysis of a Government.

Fortunately, the Bank of Canada has published several working papers from 1999 to 2011 dealing with practical debt and risk-management problems of a Government with David Bolder as main author. The main studies that have influenced the course of our research are Bolder (2006), Bolder and Liu (2007) and Bolder and Deeley (2011).
In particular, Bolder (2006) selects alternative non-macro term-structure models from the literature in order to identify a model that provides a reasonable description of interest rate dynamics for risk management purposes. In this regards, he also picks up the extended Nelson-Siegel model suggested by Diebold and Li (2006) and introduces two additional generalisations. Examining the selected models in terms of their forecast performance, their ability to capture deviations from the expectations hypothesis and their predictions in a simplified portfolio optimisation exercise, Bolder (2006) concludes that the extended Nelson-Siegel model framework provides the most appealing modelling approach under the defined criteria.
In Bolder and Liu (2007), the scope is extended by the investigation in models that provide a joint description of the macro economy and the term-structure of interest rates. Again they pick up the Nelson-Siegel motivated approaches from Bolder (2006) and incorporate macroeconomic factors in the model framework. As competitive models, they take up the concept from Ang and Piazessi (2003), who introduce a no-arbitrage joint-macro model concluding that the forecasting performance improves when no-arbitrage restrictions are imposed and macroeconomic variables are included. The models are examined by various out-of-sample forecasting tests. Similar to Bolder (2006), they conclude that the Diebold and Li (2006) motivated approaches provide the most appealing modelling alternative from a practical risk management perspective.

Finally, Bolder and Deeley (2011) provide a comprehensive overview on the debt strategy and risk management model developed by the Bank of Canada. They outline and describe the main elements of the Canadian debt-strategy model including the set of implemented stochastic joint-macro models and diagnostic tools to examine the joint dynamics of macroeconomic variables and the term-structure of interest rates. As stochastic models they have implemented not less than five joint-macro models in their debt strategy analysis. Three models are approaches that follow the extended Nelson-Siegel model suggested by Diebold and Li (2006). One uses the original mapping, the other two are generalisations using an alternative mapping, in particular exponential spline and Fourier-series motivated mappings. Moreover, a no-arbitrage observed-affine model is implemented. However, Bolder and Deeley (2011) outline that there is a reasonable amount of empirical evidence that empirical models, such as the extended Nelson-Siegel model, outperform no-arbitrage models in terms of out-of-sample forecasting.

Leveraging on the results of the working papers published by the Bank of Canada we focus our research on the extended Nelson-Siegel model suggested by Diebold and Li (2006) and follow various developments on its model specification introduced in finance literature.
In this regards, we introduce the Svensson model as well which is an extension of the Nelson-Siegel model and a popular term-structure model among central banks. The investigation on the Svensson model is motivated from the insights of de Pooter (2007), who examines several variations of the Nelson-Siegel model and concludes that more sophisticated models, such as the Svensson model, achieve better in-sample fit and out-of-sample forecasts of the term-structure of interest rates.
Moreover, we investigate in alternative estimation methodologies. This is motivated due the lack of theoretical foundation of the pre-specification of model parameters within the two-step estimation approach suggested by Diebold and Li (2006).
In this regards, we explore the non-linear model specification of the Nelson-Siegel model following de Pooter (2007) and Gilli et al. (2010), and consequently apply a heuristic optimisation approach, the Differential Evolution.
In addition, we include the developments on the extended Nelson-Siegel model introduced by Diebold et al. (2006), who reformulate the model into a state-space representation and introduce an one-step estimation approach using the Kalman filter.

The approach we use to include macroeconomic factors in a term-structure modelling framework is based on the work from Bolder and Liu (2007) and Diebold et al. (2006). In this general specifications we examine the ability of the Nelson-Siegel and Svensson models to capture the dynamics of the term-structure in our data sample and compare their in-sample fit. Another important facet enabled in this model framework is the analysis of the dynamic interactions between macroeconomic factors and the term-structure. We follow Bolder and Liu (2007) and Diebold et al. (2006), and apply the impulse response function as diagnostic tool to investigate the effects of changes in key variables on other variables in the Nelson-Siegel model.
Finally, we examine the out-of-sample forecast ability of the models in different settings. We set the original extended Nelson-Siegel model suggested by Diebold and Li (2006) as benchmark and compare it to the introduced model variations.

The data analyses and the graphical presentations of the results are performed in **R** a free software environment for statistical computing and graphics[1].

[1] R Core Team (2018).

2. The extended Nelson-Siegel model introduced by Diebold and Li (2006)

Originally the work of Nelson and Siegel (1987) focuses entirely on the problem of capturing the shapes occurring in yield curves by introducing a parametrically parsimonious model. Diebold and Li (2006) extend the Nelson-Siegel model in order to describe the term-structure dynamics of interest rates over time. We recap some basic expressions to formulate the extended Nelson-Siegel model. Therefore we combine the structure of Bolder (2006) and Diebold and Li (2006) using the notation from the former.

Let $P(t,T)$ denote the discount bond price at time t and maturity $\tau = T - t$ with $t \leq T$, and $z(t,T)$ its continuously compounded zero-coupon rate having the relationship

$$P(t,T) = e^{-z(t,T)(T-t)}, \quad \text{with } t < T. \tag{2.1}$$

The instantaneous forward rate is defined as

$$f(t,T) = \lim_{T' \to T} f(t,T,T'), \quad \text{with } T \leq T', \tag{2.2}$$

and inserting the continuously compounded forward interest rate

$$f(t,T,T') = \frac{1}{T'-T} \ln \left(\frac{P(t,T)}{P(t,T')} \right), \tag{2.3}$$

an alternative expression of the instantaneous forward rate can be derived

$$f(t,T) = -\frac{P_T(t,T)}{P(t,T)}. \tag{2.4}$$

Using (2.1) and transforming equation (2.4) a direct link between the instantaneous forward rate and the zero coupon rate can be derived

$$z(t,T) = \frac{1}{T-t} \int_t^T f(t,u)du. \tag{2.5}$$

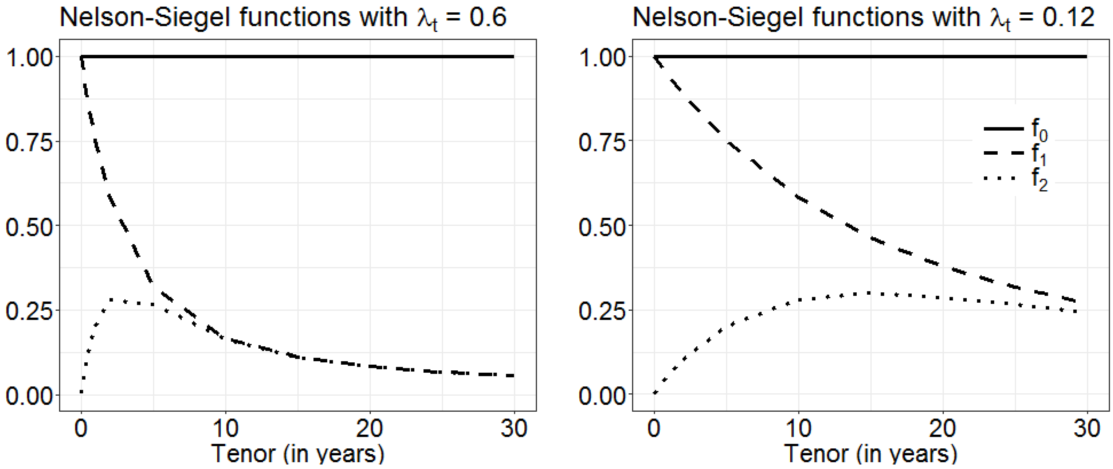
Nelson and Siegel (1987) propose in their work a specific functional form of the instantaneous forward rate

$$f(t,T) = x_0 + x_1 e^{-\lambda_t (T-t)} + x_2 \lambda_t (T-t) e^{-\lambda_t (T-t)}, \tag{2.6}$$

with the parameters $x_i \in \mathbb{R}, i = 0,1,2$ and $\lambda_t \in \mathbb{R}$. Using the relationship between $f(t,T)$ and $z(t,T)$ we can derive following expression of the zero coupon rate suggested by Diebold and Li (2006)

$$z(t,T) = x_0 + x_1 \left(\frac{1 - e^{-\lambda_t (T-t)}}{\lambda_t (T-t)} \right) + x_2 \left(\frac{1 - e^{-\lambda_t (T-t)}}{\lambda_t (T-t)} - e^{-\lambda_t (T-t)} \right). \tag{2.7}$$

Figure 2.1.: Nelson-Siegel factor loadings



The formula in (2.7) differs from the classical Nelson-Siegel approach which depends only on the tenor $\tau = T - t$ and has the following form

$$z(\tau) = \hat{x}_0 + \hat{x}_1 \frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - \hat{x}_2 e^{-\lambda_t \tau}. \tag{2.8}$$

Obviously the original Nelson-Siegel factorisation in (2.8) matches the factorisation of the extended model (2.7) with $\hat{x}_0 = x_0$, $\hat{x}_1 = x_1 + x_2$ and $\hat{x}_2 = x_2$.

Diebold and Li (2006) outline the benefit of the extension of the Nelson-Siegel model based on the fact that the original Nelson-Siegel factorisation has a similar monotonically behavior. This raises difficulties in the estimation of the coefficients $\hat{x}_i, i = 0, 1, 2$ and subsequently complicate intuitive interpretations. We will see in later sections that the non-linear model specification of the extended Nelson-Siegel model still implies numerical problems in estimating robust values for the parameters x_i, $i = 0, 1, 2$.

However, Diebold and Li (2006) introduce revealing observations on the characteristics of the coefficients x_1, x_2 and x_3 in the extended Nelson-Siegel model. Apparently, the parameter λ_t sets the exponential decay rate of the respective functions, hereinafter referred to as factor loadings,

$$f_0(\tau) = 1, f_1(\tau) = \frac{1 - e^{-\lambda_t \tau}}{\lambda_t y}, f_2(\tau) = \frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau}. \tag{2.9}$$

Figure 2.1 plots the functions $f_i, i = 0, 1, 2$ with different values for the decay parameter λ_t over the tenor $\tau = T - t$ of 30 years. It illustrates that small values of λ_t produce slow decay whereas large values imply fast decay of the function values. The decay parameter λ_t also governs the tenor value where f_2 achieves its maximum. Furthermore, Figure 2.1 reveals that f_0 has a consistent impact over all tenors, f_1 has a strong effect at short tenors which decreases at longer tenors and f_2 has an over-proportional impact on the middle range of the tenors. Leveraging on the insights of characteristics of the functions, Diebold and Li (2006) derive the effects on the shape of the term-structure caused by changes in the coefficients $x_i, i = 0, 1, 2$ as follows:

- x_0 is the long-term factor, changes create parallel shifts up or down of the term-structure curve.
- x_1 is the short-term factor, changes create a steepening or flattening of the term-structure curve.
- x_2 is the medium-term factor, changes create a decreasing or increasing of the curvature of the term-structure curve.

The very useful insight made by Diebold and Li (2006) is that the coefficients may be interpreted as the state variables level l_t, slope s_t, and curvature c_t. They relate this important result to other finance literature, such as Litterman and Scheinkman (1991), which indicate that the dynamics of term-structure curves can be described by only a few latent state factors.

Consequently, Diebold and Li (2006) re-interpret the classical Nelson-Siegel model as a dynamic model describing the dynamics of the term-structure of interest rates over time. In particular, equation (2.7) is reformulated by time-varying state factors l_t, s_t, c_t and decay parameter λ_t,

$$z_t(\tau) = l_t + s_t \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} \right) + c_t \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau} \right). \tag{2.10}$$

The parameters can be estimated simultaneously at one point in time t using non-linear least squares or other non-linear optimisation techniques. However, Diebold and Li (2006) suggest to fix λ_t at a pre-specified value over all t and subsequently apply ordinary least squares to estimate l_t, s_t and c_t. As a consequence one obtain a time series of estimated state variables over time. The dynamics of the obtained time-series of the state vector $\{X_t\} = (l_t, s_t, c_t)'$ can be modeled by univariate autoregressive models (AR) respectively multivariate vector autoregression models (VAR). Obviously, the suggested estimation approach is clearly separated in two steps and therefore it is also referred to as two-step estimation approach in the finance literature.

Bolder (2006) picks up the concept of the extended Nelson-Siegel model and introduce to the common factor loadings in (2.9) two generalizations using exponential-spline and Fourier-series basis functions. He finds that the generalisations are competitive to the original setting of the extended Nelson-Siegel model, especially in terms of forecasting the term-structure of interest rates. Nevertheless, in Bolder and Liu (2007) they outline that if they wish to select a single model, their first choice would be the extended Nelson-Siegel model as suggested by Diebold and Li (2006).
Concerning this statement and the subsequent introduction of the re-formulation from Diebold et al. (2006) and the Svensson model where both are building up on the basis functions in (2.9) we confine ourselves on the common factor loadings.

3. Variations of the extended Nelson-Siegel model and estimation methodologies

Initially, we have investigated on the enhancement of the extended Nelson-Siegel model by macroeconomic information and the incorporation of such a joint-macro model into debt strategy and risk-management analysis. However, the knowledge gained from this work and additional finance literature has indicated a benefit of investigating in further variations within the Nelson-Siegel class of term-structure models.

Fortunately, de Pooter (2007) collects the main models within the Nelson-Siegel class of the term-structure models, such as the Svensson model, elaborates their characteristics and examines their ability in terms of forecasting the term-structure of interest rates. In general, the variations of the extended three-factor Nelson-Siegel model can be outlined by the inclusion of additional slope or curvature factors as well as additional decay parameters λ_t^i. The studies from de Pooter (2007) indicate that more flexible model variations achieve better in-sample fit of the term-structure of interest rates and more importantly improve the out-of-sample forecast performance as well. The best forecast results have been achieved with four-factor models, in particular the Björk and Christensen (1999) and the Svensson (1994) model, which include an additional factor to the three-factor Nelson-Siegel model. Additionally, de Pooter (2007) states that the Bank of International (BIS) has evaluated in 2005 that a large part of the central banks use either the Nelson-Siegel model or the Svensson model. Therefore, we confine ourselves on introducing the Svensson model.

Moreover, we have delved into further estimation methodologies applied in the class of Nelson-Siegel term-structure models.
At first we introduce the two-step estimation approach as suggested by Diebold and Li (2006). In this regards, we set pre-specified λ values in the Nelson-Siegel and Svensson model specifications and apply ordinary least squares in order to estimate the state factors. However, concerning the lack of a theoretical foundation for the pre-specification of the λ parameters we extend the two-step estimation approach by a non-linear optimisation approach. The application of non-linear estimation approaches permits the estimation of the state variables l_t, s_t and c_t and the decay parameter λ_t simultaneously at one point in time t. Consistent with finance literature we have faced numerical problems in estimating robust parameters. In this regards, Gilli et al. (2010) have analysed the calibration of the Nelson-Siegel and Svensson model in detail and introduce an alternative optimisation heuristic, the Differential Evolution (DE). We summarise the identified obstacles in the model specifications of the Nelson-Siegel and Svensson model following Gilli et al. (2010) and introduce the Differential Evolution method in our model framework.

Furthermore, we introduce the developments from Diebold et al. (2006), who reformulate the extended Nelson-Siegel model into its state-space representation and suggest a one-step estimation approach.

In the remaining part of this section we introduce the Svensson model. Moreover, we establish the extended estimation methodology by the two step estimation approach with pre-specified and variable decay parameters. Subsequently, we introduce the one-step estimation approach suggested by Diebold et al. (2006).

3.1. Svensson four-factor model

As outlined above the Svensson (1994) model is with the extended Nelson-Siegel model one of the popular term-structure models among central banks. It is a more sophisticated model as it is extended by an additional curvature factor c_t^2 including a second decay parameter λ_t^2,

$$z_t(\tau_i) = l_t \cdot \underbrace{1}_{f_0} + s_t \cdot \underbrace{\left(\frac{1-e^{-\lambda_t^1 \cdot \tau_i}}{\lambda_t^1 \cdot \tau_i}\right)}_{f_1} + c_t^1 \cdot \underbrace{\left(\frac{1-e^{-\lambda_t^1 \cdot \tau_i}}{\lambda_t^1 \cdot \tau_i} - e^{-\lambda_t^1 \cdot \tau_i}\right)}_{f_2} + c_t^2 \cdot \underbrace{\left(\frac{1-e^{-\lambda_t^2 \cdot \tau_i}}{\lambda_t^2 \cdot \tau_i} - e^{-\lambda_t^2 \cdot \tau_i}\right)}_{f_3}. \quad (3.1)$$

Figure 3.1 presents the factor loadings of the Svensson model with different values of the decay parameters λ_t^i over tenor τ. de Pooter (2007) highlights numerical difficulties in the estimation of the state variables $X_t = (l_t, s_t, c_t^1, c_t^2)'$ using non-linear optimisation methods. Especially, in the case when the decay parameters assume similar values and the model reduces to the three-factor Nelson-Siegel model - right plot of Figure 3.1. Then optimisation methods have problems to individually estimate the curvature state variables c_t^1 and c_t^2. de Pooter (2007) approaches this problem by introducing an adjusted second curvature loading,

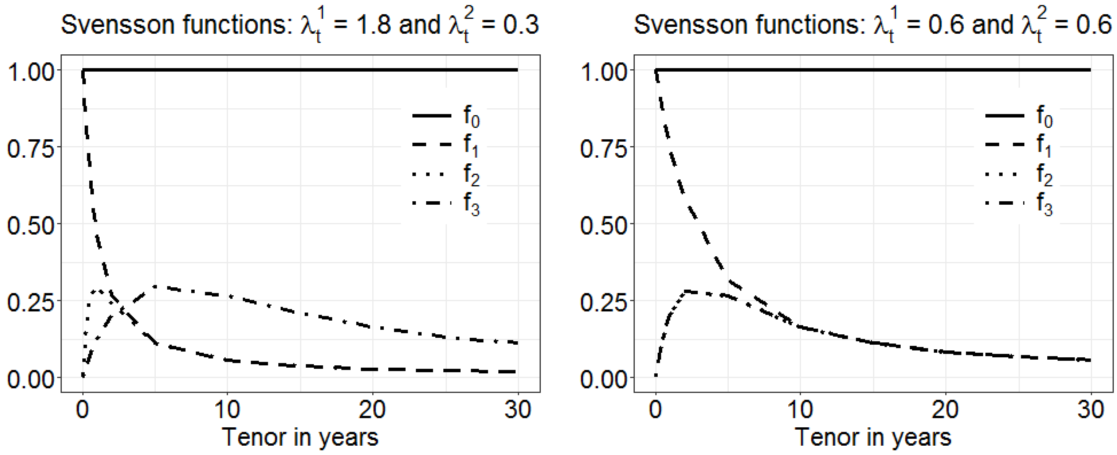
$$f_3 = \left(\frac{1-e^{-\lambda_t^2 \cdot \tau_i}}{\lambda_t^2 \cdot \tau_i} - e^{-2 \cdot \lambda_t^2 \cdot \tau_i}\right). \quad (3.2)$$

We have experimented with this adjustment in our model framework but have not identified significant superior results in terms of in-sample fit and out-of-sample forecasting to the original Svensson model. In addition, we have not found a further application of this model adjustment in the finance literature. Therefore, we focus in this paper on the results achieved with the original Svensson model.

3.2. Estimation methodologies

Various estimation methodologies have been introduced in the framework of the Nelson-Siegel class of term-structure models by finance literature. The most straightforward approach, suggested in Diebold and Li (2006), is to initially fix λ_t over all t and subsequently

Figure 3.1.: Svensson factor loadings



apply ordinary least squares regression in order to estimate the state variables. In numerically more challenging approaches, the state factors and the decay parameters are estimated simultaneously, at one point in time t, using non-linear least squares or other non-linear optimisation techniques.
In both approaches, the dynamics of the obtained time series of the state variables are described subsequently by autoregressive or vector-autoregressive models. Therefore, this estimation approach is clearly separated in two steps. Diebold et al. (2006) highlight that the information on the uncertainty associated to the observed interest rates is not acknowledged in the second step estimating the dynamics of the state factors. In this regards, they reformulate the extended Nelson-Siegel framework into a state-space model and introduce a one-step estimation approach.

In the remaining section we introduce the estimation procedures suggested in finance literature in detail and describe their implementation in our model framework.

3.2.1. Two-step estimation approach with fixed decay parameters

Diebold and Li (2006) suggest in the first introduction of the extended Nelson-Siegel model the two-step estimation approach. They propose to initially fix λ_t over all t and subsequently apply ordinary least squares regression in order to estimate the state variables l_t, s_t and c_t. In the Nelson-Siegel model specification, they outline that the fixed λ_t parameter is commonly determined in a way that the factor loading f_2 of the curvature c_t achieves its maximum between two- or three-year tenors. We follow this approach and initially set $\lambda = 0.5978$ for all t. Therefore, the linked curvature factor loading f_2 achieves its maximum at a three year tenor[1].

In the Svensson model we need to fix two decay parameters, namely λ_t^1 and λ_t^2. In finance literature we have not found any indication on plausible pre-specified values for the decay

[1]In contrast, Diebold and Li (2006) pre-specify λ over all t in the way that the curvature factor loading reaches its maximum at 30 months.

parameters in the Svensson model. However, based on the basic assumption of Diebold and Li (2006) and own observations on the results of the more advanced estimation techniques we have pre-specified the decay parameters as follows,

$$\lambda_t^1 = 1.8023, \quad \lambda_t^2 = 0.5978. \tag{3.3}$$

The parametrisation can be interpreted in the way that the curvature factor loading f_2 achieve its maximum at 1 year tenor, whereas the curvature factor loading f_3 achieve its maximum at 3 years tenor. In this regards, we would like to highlight that similar to the suggestion in Diebold and Li (2006) for the Nelson-Siegel specification the assumption on the pre-specified values of the decay parameters is not based on robust theoretical foundation. However, we have experienced that the results of the Svensson model with fixed decay parameters are competitive to the other model specifications in terms of the in-sample fit and out-of-sample forecasting of the term-structure of interest rates.

Given pre-specified values of the decay parameters, we can calculate the values of the factor loadings for both model specifications and subsequently apply ordinary least squares at each point in time t obtaining time series of the state variables. In our model framework the ordinary least squares is applied using the R-function `lm()` from the standard package **stats** of **R**[2].
In the in-sample fit and out-of-sample forecast analysis the two-step estimation approach with fixed decay parameters will be referred to as `2-step fix` method.

3.2.2. Two-step estimation approach with variable decay parameters

In literature there are different non-linear estimation approaches introduced for the Nelson-Siegel and Svensson model all facing numerical challenges in the estimation of the model parameters. de Pooter (2007) outlines that the non-linear specification of the models seems to cause numerical difficulties for optimisation methods in identifying robust estimates, which might result in extreme values of the state variables. However, an estimation method has to fit not only the term-structure of interest rates well, but also has to identify robust parameters over time to allow a reasonable modelling of their evolution with the final objective to generate plausible interest rate forecasts.
Gilli et al. (2010) analyse the calibration of the Nelson-Siegel and Svensson model in detail identifying the specifications of the models that imply the numerical difficulties for non-linear optimisation methods. They argue that the optimisation problem in the Nelson-Siegel and Svensson model is not convex and has multiple local optima, thereby repeating standard optimisation techniques with various randomly drawn starting values the estimated state variables vary widely from one estimation run to another. Moreover, they identify multicollinearity among the factor loadings of the Nelson-Siegel and the Svensson model for many different ranges of the decay parameters. This causes difficulties in uniquely identifying parameter estimations which can result in extreme values of the state variables. The multicollinearity inherent in the Svensson model specification is most evident examining the factor loadings in the case that λ_t^1 and λ_t^2 are (roughly) equal - see right plot of Figure

[2] R Core Team (2018).

3.1. These characteristics in the Nelson-Siegel and Svensson model specifications have been identified by de Pooter (2007) as well. In both studies, the multicollinearity problem is approached by imposing restrictions on the ranges of the decay parameter values. Moreover, Gilli et al. (2010) test an optimisation heuristic, in particular the Differential Evolution (DE), concluding that the DE method is more appropriate than a traditional optimisation technique based on the gradient[3].

Consequently, we extend the two step estimation approach by the Differential Evolution (DE) method based on the studies from Gilli et al. (2010). The DE method is implemented in our model framework using the **R**-package **NMOF** described in Schumann (2019) that accompanies the book Gilli et al. (2019). Consistent with the underlying literature we have experienced numerical difficulties with the more flexible estimation approach. The numerical problems have caused the occurrence of extreme values in the time series of the decay parameters and the state variables. In our model framework, extreme values in the estimated time series of the state factors cause difficulties in their economical interpretation and in the modelling of their dynamics and interactions. As outlined above, the collinearity problem has been approached in de Pooter (2007) and Gilli et al. (2010) by imposing restrictions on the value ranges of the decay parameters.
In particular, de Pooter (2007) imposes general range restrictions on the decay parameters λ_t^i of all examined models by limiting the curvature factor loadings to achieve their maximum only for tenors between one and five years. On the contrary, Gilli et al. (2010) determines general value ranges for λ_t^i on which the factor loadings should result in acceptable correlations. We have combined and adapted the described limitations increasing the tenor range for the curvature hump due to longer maturities in our data sample[4], and due to the subsequent application on the Svensson model. Therefore, we define the following general restrictions on the decay parameters: the curvature factor loadings of both models - i.e. f_2 and f_3 - are allowed to achieve their maximum for the tenors from 0.75 years to 7 years. The respective value ranges of the decay parameters are $[0.2561, 2.3753]$.

The specific collinearity problem inherent in the Svensson model when λ_t^1 and λ_t^2 are (roughly) equal has been approached by additional restrictions on the decay parameters. In particular, de Pooter (2007) limits λ_t^2 in the way that the maximum of its curvature factor loading f_3 is at least twelve months shorter than the tenor value of the maximum of the first curvature factor loading f_2. On the other hand, Gilli et al. (2010) impose limitations on the Svensson model by segregating the value ranges of the two decay parameters λ_t^i. In particular, they define the following restrictions in their DE parametrisation:

$$0 \leq \lambda_t^1 \leq 2.5, \quad 2.5 \leq \lambda_t^2 \leq 5.5. \tag{3.4}$$

This set up aims to limit the correlation among the factor loadings to an acceptable magnitude. As we have implemented the DE method we follow the approach from Gilli et al. (2010). Translating their methodology into our model framework we receive the following

[3] In contrast, de Pooter (2007) uses non-linear least squares.
[4] Up to 30 years instead of 10 years in de Pooter (2007)

parametrisation for λ_t^i[5]:

$$0.4 \leq \lambda_t^1, \quad 0.18 \leq \lambda_t^2 \leq 0.4. \tag{3.5}$$

However, we have refined the parametrisation, considering the general restrictions already imposed above, as follows: λ_t^1 is allowed to vary over the values which maximises the curvature factor loading from 0.75 year to 3 years and λ_t^2 respectively to maximise its curvature loading from 3 years to 7 years. This limitation implies the following DE parametrisation:

$$0.5987 \leq \lambda_t^1 \leq 2.3753, \quad 0.2561 \leq \lambda_t^2 \leq 0.5987. \tag{3.6}$$

The role of the decay parameters λ_t^1 and λ_t^2 in the Svensson model are interchangeable in terms of the curvature factor loadings as already highlighted by de Pooter (2007)[6].
We have observed that the restrictions have a marginal impact on the goodness of the in-sample fit in comparison to parametrisations with unrestricted decay parameters. But on the other hand a significant positive effect on the robustness of the estimated state variables and consequently on the out-of-sample forecasting performance of the models.
Investigations on different restrictions on the decay parameter have strengthened the assertions of the positive effect of the approach[7]. However, only marginal differences have been observed among the restriction variations therefore we have carried on our studies with the described limitations. Nevertheless, this has been a first try to impose restrictions on the decay parameters based on the work of de Pooter (2007), Gilli et al. (2010) and own observations. Further investigation might be rewarding.

The explicit usage of the DE-method in **R** is comprehensively described in Gilli et al. (2010) and Schumann (2019). The implemented DE method permits the specification of lower and upper boundaries for the model parameters to be estimated. Therefore, we can easily implement the limitations as defined above. In addition, the implemented DE-method requires the Nelson-Siegel and Svensson models as functions having the model parameters as arguments, and the objective function which has to be minimized. The first step is done straightforward in **R** by defining the model specification in the equations (2.10) and (3.1) as functions. As we will compare the in-sample fit of the models in terms of the root-mean-squared error over the term-structure of interest rates we have defined this measure as objective function in the DE-method. The measure is defined in equation (5.1).
The two-step estimation approach with variable decay parameters will be referred to as `2-step var` method.

[5] In their model framework the curvature loading is defined as follows: $\left(\frac{1-e^{-\tau/\lambda_t^i}}{\tau/\lambda_t^i} - e^{-\tau/\lambda_t^i}\right)$.

[6] On the other hand, the λ_t^1 affects the slope factor loading f_1 as well. Therefore, we have experimented with reversed restrictions on the λ_t^i parameters. However, we have not identified significant differences in the general performance of the models in terms of in-sample fit and out-of-sample forecasting. As a consequence we present solely the results of the defined λ-parametrisation.

[7] E.g. restrictions on λ value ranges that have been obtained by deriving confidence intervals from the time series of λ_t^i obtained by unrestricted model parametrisations.

3.2.3. One-step estimation approach

Diebold et al. (2006) reformulate the extended Nelson-Siegel model with the main goal to integrate macroeconomic factors and to analyse the dynamic interactions between the macro economy and the term-structure of interest rates. In this regards, they introduce a state-space representation of the extended Nelson-Siegel model. To estimate the model they suggest the Kalman filter which allows the simultaneous fitting of the observed interest rates and the estimation of the underlying dynamics of the state variables. Therefore, this estimation approach is referred to as one-step estimation approach. The estimates of the model parameters are calculated using the log-likelihood of the observed interest rates derived by the Kalman filter. Moreover, the Kalman filter delivers optimal filtered and smoothed dynamics of the state variables. Diebold et al. (2006) concludes that they clearly prefer the one-step estimation approach to the two-step estimation approach. They argue that the Kalman filter uses information from the observed interest rates in the estimation of all parameters which produces correct inference via standard theory. In contrast, they outline that the two-step estimation approach suffers from the fact that the parameter estimation and signal extraction uncertainty associated with the first step is not acknowledged in the second step.
In this regards, we would like to highlight that while Diebold et al. (2006) prefer the one-step estimation approach, Bolder adhere to the two-step estimation approach in Bolder (2006) and Bolder and Liu (2007). In particular, in Bolder and Liu (2007) it is stated that they have experimented with the Kalman-filter approach but obtained superior results with the two-step estimation method with respect to their purpose.

The state space representation of the extended Nelson-Siegel model as suggested by Diebold et al. (2006) is defined as follows. The dynamics of the state variables $\{X_t\} = (l_t, s_t, c_t)'$ are described in the state equation,

$$\begin{pmatrix} l_t - \mu_l \\ s_t - \mu_s \\ c_t - \mu_c \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} l_{t-1} - \mu_l \\ s_{t-1} - \mu_s \\ c_{t-1} - \mu_c \end{pmatrix} + \begin{pmatrix} \eta_t(l_t) \\ \eta_t(s_t) \\ \eta_t(c_t) \end{pmatrix}, \tag{3.7}$$

for $t = 1, \ldots, T$. The measurement equation links the observed interest rates z_t with the unobservable state variables,

$$\begin{pmatrix} z_t(\tau_1) \\ z_t(\tau_2) \\ \vdots \\ z_t(\tau_N) \end{pmatrix} = \begin{pmatrix} 1 & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} & \frac{1-e^{-\lambda\tau_1}}{\lambda\tau_1} - e^{-\lambda\tau_1} \\ 1 & \frac{1-e^{-\lambda\tau_2}}{\lambda\tau_2} & \frac{1-e^{-\lambda\tau_2}}{\lambda\tau_2} - e^{-\lambda\tau_2} \\ \vdots & \vdots & \vdots \\ 1 & \frac{1-e^{-\lambda\tau_N}}{\lambda\tau_N} & \frac{1-e^{-\lambda\tau_N}}{\lambda\tau_N} - e^{-\lambda\tau_N} \end{pmatrix} \cdot \begin{pmatrix} l_t \\ s_t \\ c_t \end{pmatrix} \begin{pmatrix} \epsilon_t(\tau_1) \\ \epsilon_t(\tau_2) \\ \vdots \\ \epsilon_t(\tau_N) \end{pmatrix}, \tag{3.8}$$

for $t = 1, \ldots, T$. The state-space representation is straightforward formulated in vector/-matrix notation,

$$(X_t - \mu) = A(X_{t-1} - \mu) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q), \tag{3.9}$$

$$z_t = \Lambda X_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, H). \tag{3.10}$$

The white noise state (H) and measurement (Q) disturbances are assumed to be orthogonal to each other and to the initial state vector. In addition, the covariance matrix H is assumed to be diagonal whereas the covariance matrix Q is assumed to be non-diagonal.
The log-likelihood of the state space representation in (3.8) - (3.10) is given by

$$\ell(\theta) = \log p(z_0, \ldots, z_T|\theta) = \\ = -\frac{(T+1)N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=0}^{T}(\log|F_{t|t-1}| + e'_{t|t-1}F^{-1}_{t|t-1}e_{t|t-1}), \quad (3.11)$$

where $\theta = \{A, \mu, Q, H, \lambda\}$ is the set of parameters to be estimated. $F_{t|t-1} = \mathbf{E}[e_{t|t-1}e'_{t|t-1}]$ is the conditional covariance matrix of the prediction errors $e_{t|t-1} = z_t - z_{t|t-1}$, where $z_{t|t-1}$ is the vector of interest rate forecasts given information up to time $t-1$ and z_t are the observed interest rates at time t.

In general, the implementation of the Kalman filter algorithm in programming language is straightforward given a state space model. However, by recursively iterating over a big data sample of observed interest rates the numerical stability and speed of the implemented Kalman filter is important. Tusell (2011) reviews five alternative **R**-packages supporting state-space estimation via Kalman filtering. In this paper, the features of the packages are introduced and their abilities in terms of speed and parameter estimation using maximum-likelihood estimation is examined. We have reviewed the five packages and have found two packages most appropriate for our purposes, **FKF** (Luethi et al. (2018) and **dlm** (Petris (2010)), whereby the later accompanies the book Petris et al. (2009). Both packages provide functions for performing Kalman filtering on a given state-space model and initial values, and returning the respective value of the log-likelihood function. The state-space representations supported by the packages are various, however, the relevant set-up for our purpose can be generalized as follows,

$$S_t = AS_{t-1} + u_t, \quad u_t \sim \mathcal{N}(0, Q), \quad (3.12)$$

$$y_t = CS_t + v_t, \quad v_t \sim \mathcal{N}(0, H), \quad (3.13)$$

where S_t is the state vector, A the transition matrix, C the measurement matrix and y_t the observations. Defining the latent state vector as mean-adjusted state variables $S_t = X_t - \mu$, it becomes obvious that the transition matrices A in equation (3.9) and (3.12) are ident. Moreover, inserting the state vector into the measurement equation (3.10) we get,

$$z_t - \Lambda\mu = y_t = \Lambda S_t + v_t, \quad v_t \sim \mathcal{N}(0, H). \quad (3.14)$$

Consequently, we need to use the mean-adjusted interest rates as observations $y_t = z_t - \Lambda$ in order to reformulate the state-space presentation in (3.9)-(3.10) that it is supported by the **R** set-up.
Given the state-space equation and initial values, the **R**-packages **dlm** and **FKF** provide Kalman filter functions and the respective log-likelihood calculations. Therefore, we only need to combine the `mle()` function of the standard **stats4** package of **R** and the log-likelihood results of the Kalman filter functions in order to find the minimum of the

negative log-likelihood function. The `mle()` function uses the standard function `optim()` from **stats** package that provides several optimisation methods for performing minimisation. In particular, we have applied the BFGS-method which is a quasi-Newton method based on the well-known Broyden-Fletcher-Goldfarb-Shanno algorithm[8]. In certain model specifications we have experienced extreme or negative values in the decay parameters during the optimisation run causing the termination of the maximum likelihood estimation. Therefore, we have used additionally for these cases the L-BFGS-B-method which is a limited memory modification of the BFGS algorithm allowing box constraints on the parameters to be estimated[9].
In order to assess the adequateness of our state-space model specification in **R** we have applied it on the yield dataset from Diebold et al. (2006). In this regards, there is a detailed description for the implementation of the one-step Kalman filter estimation suggested by Diebold et al. (2006) for MATLAB[10]. In order to identify plausible starting values of the parameters to be estimated the two-step estimation approach is applied as suggested by Diebold and Li (2006). Applying our model set-up in **R** on the yield data from Diebold et al. (2006) we obtain identical results in terms of parameter estimates and in-sample fit of the yields.

In general, we have achieved consistent results with both packages in terms of parameter estimates and log-likelihood values. Both packages can be used to receive smoothed time series of the state vector as well. However, the Kalman filter in **dlm** provides robust numerical stability as a form of square root filter is used which propagates factors of the singular value decompositions as outlined by Tusell (2011). On the contrary, the package **FKF** provides a fast and flexible implementation of the Kalman filter that significantly outperforms the other package in terms of speed. In addition, in comparison to the other packages it allows intercept vectors in the state-space representation. Using this feature we can reformulate the state space presentation of the extended Nelson-Siegel model following de Pooter (2007),

$$X_t = \mu + AX_{t-1} + u_t, \quad u_t \sim \mathcal{N}(0, Q), \tag{3.15}$$

$$z_t = CX_t + v_t, \quad v_t \sim \mathcal{N}(0, H), \tag{3.16}$$

where μ is interpreted as intercept vector. In this set-up we do not need to adjust the interest rates by the mean. However, comparing the results to the original state space representation we have identified no significant difference in terms of calculated log-likelihood values and obtained smoothed state vector estimates.

The above described approach can be easily applied on the Svensson model. Therefore, only the state-space model have to be adapted by adding the additional curvature factor c_t^2 to the state vector as well as including the additional factor loading f_2 to the measurement matrix.
Hereinafter, the one-step estimation approach will be referred to as `1-step` method.

[8] https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm
[9] https://en.wikipedia.org/wiki/Limited-memory_BFGS
[10] https://www.mathworks.com/help/econ/examples/using-the-kalman-filter-to-estimate-and-forecast-the-diebold-li-model.html

3.2.4. One-step-two-step estimation approach

Recap that we have introduced further estimation methodologies due to the lack of theoretical background on the pre-specification of the decay parameters within the two-step estimation approach suggested by Diebold et al. (2006). Moreover, as outlined in previous section 3.2.3, Bolder and Liu (2007) prefer the two-step estimation approach, therefore, we combine the one-step and two-step estimation approach. In particular, we use the estimates of decay parameters obtained by the one-step estimation approach for subsequently applying the two-step estimation approach with fixed decay parameters as described in section 3.2.1.
This approach will be referred to as `1-2-step` method.

4. Extension of the Nelson-Siegel Class of term-structure models with macroeconomic factors

Main focus of the work has been the research of joint-macro term-structure models that are appropriate and convenient for debt-strategy and risk management analysis of a Government. The research of the finance literature has brought us to working papers of the Bank of Canada driven by David Bolder as outlined in section 1. Several papers examine alternative term-structure models from a debt-strategy and risk management analysis perspective. Based on the preference of David Bolder on models motivated by the Diebold and Li (2006) approach we have focused our analysis on the extended Nelson-Siegel model and close further developments.
In this section we describe the used macroeconomic data and two approaches for incorporating macroeconomic factors into the extended Nelson-Siegel model that are suggested by Bolder and Liu (2007) and Diebold et al. (2006). The different frameworks are motivated by the preferences of the authors in regards to the used estimation approach. As we have outlined in previous section 3.2.3, while Bolder and Liu (2007) adhere to the two-step estimation approach, Diebold et al. (2006) reformulate the extended Nelson-Siegel model into state-space representation and introduce a one-step estimation approach. The main objective of their work has been the incorporation of macroeconomic factors in order to formulate a joint-macro model.

4.1. Macroeconomic data

The macroeconomic factors mostly used in joint macroeconomic and term-structure modelling are inflation, economic growth factors, and monetary policy instruments. Diebold et al. (2006) use manufacturing capacity utilisation as level for real economic activity, the federal funds rate as monetary policy instrument and the annual price inflation as inflation rate. Bolder and Liu (2007) use output gap, annual inflation and a monetary policy rate and extend the model framework in Bolder and Deeley (2011) by including the growth in potential output and total consumer price index as exogenous factors. Reviewing the set of macroeconomic factors in the European Economic Area we select the annual inflation i_t and output gap o_t rate of the Economic and Monetary Union of the European Union (EMU). The monetary policy instrument in our model framework is set by the European overnight rate m_t (EONIA) which is driven by the European Central Bank (ECB) policy rate[1].

[1]EONIA (Euro OverNight Index Average) is the interest rate at which banks of sound financial standing in the European Union (EU) and European Free Trade Area (EFTA) countries lend funds in the interbank

Figure 4.1.: Macroeconomic factors

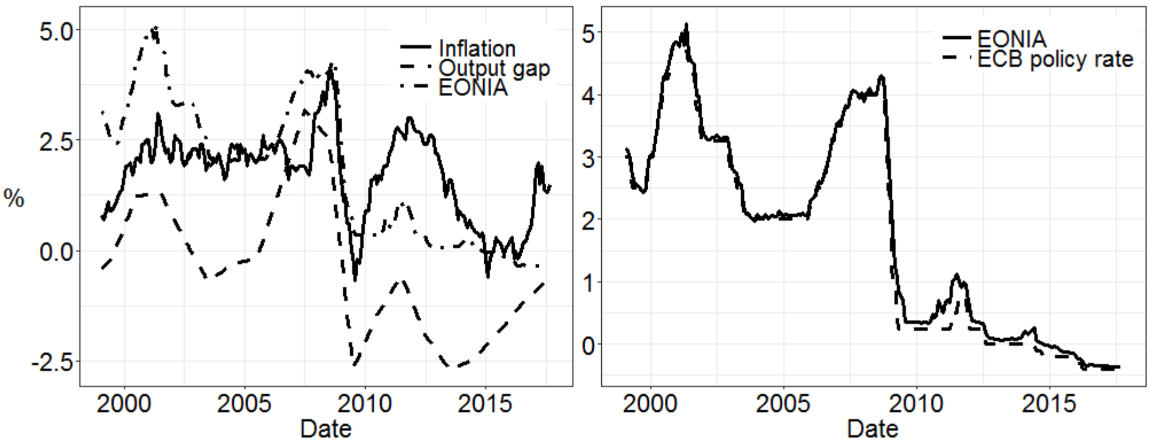


Figure 4.1 displays the data sample of the macroeconomic variables. The inflation rates are on a monthly basis, but the output gap rate is on a yearly basis. As a consequence we need to interpolate the output gap rate to derive values on a monthly basis[2]. The overnight rate EONIA is on daily basis, therefore we use the average within a month in our model framework. The right plot in 4.1 demonstrates that the EONIA rate is driven by the ECB policy rate. It is therefore reasonable to use it as the monetary policy instrument in our approach. In this regards it has to be noted that the methodology on the EONIA rate has been changed by ECB. Since October 2019 the euro short-term rate (€STR), which reflects the wholesale euro unsecured overnight borrowing costs of banks located in the euro area, and the new EONIA are published on a daily basis. It is recommended by the working group on euro risk-free rates that market participants gradually replace the EONIA with €STR making the latter to their standard reference rate[3]. The macroeconomic data has been made available for this research by the Austrian Treasury (Österreichische Bundesfinanzierungsagentur)[4].

4.2. Joint-macro term-structure model following Bolder and Liu (2007)

The extended Nelson-Siegel model suggested by Diebold and Li (2006) is easily formulated as joint-macro model. The time series of the state variables X_t obtained in the first step of the two-step estimation approach is enlarged with the macroeconomic variables inflation i_t, output gap o_t and monetary policy instrument m_t ,

$$\{\bar{X}_t\} = (l_t, s_t, c_t, i_t, o_t, m_t)'.$$

money market in euro - see https://www.emmi-benchmarks.eu/euribor-eonia-org/about-eonia.html.

[2]A linear approach is used.

[3]For more details see https://www.ecb.europa.eu/stats/financial_markets_and_interest_rates/euro_short-term_rate/html/eurostr_overview.en.html.

[4]https://www.oebfa.at/en/

Obviously, in this approach there is no direct link between the macroeconomic factors and the interest rates. Instead, the macroeconomic variables assist in the description of the dynamics of the state variables by applying a stochastic process on the enlarged time series $\{\tilde{X}_t\}$[5]. The straightforward approach to model the dynamics and interactions of the state variables and macroeconomic factors is to use a vector autoregression model (VAR).
This approach is easily applied on the Svensson model by enlarging the respective time series of the estimated state vectors,

$$\{\bar{X}_t\} = (l_t, s_t, c_t^1, c_t^2, i_t, o_t, m_t)'.$$

In the examination of the joint-macro models Bolder and Liu (2007) have identified that constant-parameter assumption on the VAR specifications is not ideal. They argue that facing different economic regimes in the macroeconomic and interest rates data, unadjusted parametrisation of the VAR models using the entire data over all regimes would be likely unreasonable. The main concern is that the forward looking description of the interest rates does not reflect the current economic regime at the end of the data but rather a weighted average of the economic regimes occurring in the data. In this regards, Bolder and Liu (2007) approach this problem by incorporating time-varying parameters in the VAR models. In particular, the intercept vectors are linked to exogenously imposed economic regimes[6]. Depending on the current regime in the macroeconomic factors the intercept vector of the VAR specification changes. Bolder and Liu (2007) concludes that permitting model parameters to vary-over-time improves the forecast performance of a term-structure model.
In order to investigate the effects of time-varying parameters in the VAR specification we will apply a simplified approach in the forecast analysis.

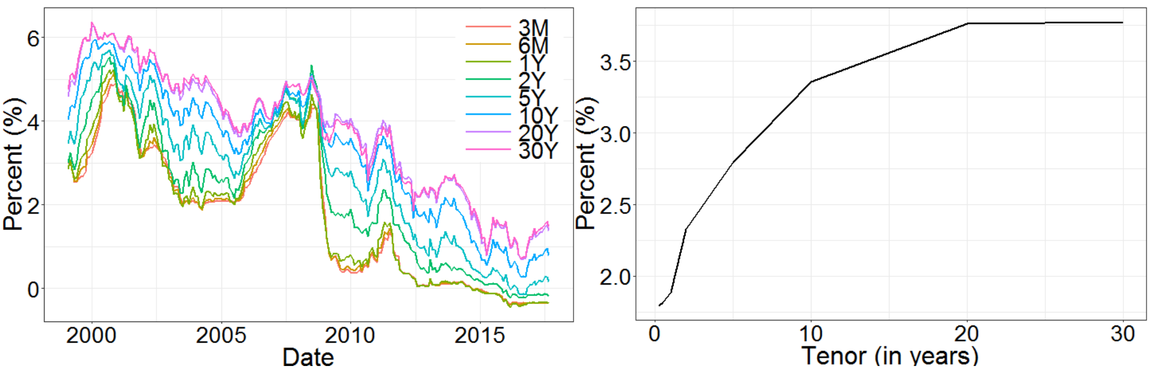
4.3. Joint-macro term-structure model following Diebold et al. (2006)

The main objective in Diebold et al. (2006) has been a formulation of a joint-macro model that provides characterisation of the dynamic interactions among macroeconomic key figures and the term-structure of interest rates. Therefore, they have reformulated the extended Nelson-Siegel model into a state-space model as described in the previous section 3.2.3. In order to characterise the interlinks among the state variable and the macro economy they included three macroeconomic factors - the set of macro variables is outlined in 4.1. The incorporation of the macro variables into the Nelson-Siegel state space model is again straightforward by enlarging the state vector $\{\bar{X}_t\} = (l_t, s_t, c_t, i_t, o_t, m_t)'$ in equations (3.7)-(3.10) and appropriately adapting the dimensions of the respective matrices and of

[5]In contrast, the joint macroeconomic and term-structure model in Ang and Piazessi (2003) links the interest rates and macroeconomic factors directly. However, as we have outlined in section 1, Bolder and Liu (2007) examines and compares this model to joint-macro models motivated by the extended Nelson-Siegel model and concluded superiority of the Nelson-Siegel approach in terms of out-of-sample forecasting.

[6]The regimes used by Bolder and Liu (2007) are identified by Demers (2003) in the Canadian inflation and output gap rate.

Figure 4.2.: Swap Rates



(a) Swap Rates Data Sample (b) Average Swap Rates Curve

the mean vector. The measurement matrix is enlarged by three additional columns inserting zero values. Therefore, the macroeconomic variables have no direct link to the observed interest rates similar to the set-up defined by Bolder and Liu (2007).
However, the application of the Kalman filter on the extended state-space representation including macroeconomic factors is not straightforward. The state variables in (3.7) - (3.10), such as in this case of the level l_t, slope s_t and curvature c_t, are in general not observable and derived in a state-space model based on initial values of the states and given observations, in our case interest rates. The unobserved state variables are linked to the observed interest rates by the measurement equations in (3.8) and (3.10) and are iteratively updated by the Kalman filter given the observations.
The macroeconomic variables on the other hand are observed variables providing information on the current macro economy regime. The state-space model implemented in R, as introduced in section 3.2.3, assumes that the state variables are unobserved variables that have to be filtered or smoothed given the observed interest rates. We have not found a straightforward solution to incorporate the macroeconomic variables as observed state variables. Therefore, we leave this extension of the state-space model open for other research work. Nevertheless, the estimated state variables and model parameters estimated in the non-macro setting of the state-space model will be used for joint-macro framework as suggested by Bolder and Liu (2007) and described in the previous section.

4.4. Interest rate data

The Nelson-Siegel model is generally applied on yield curves or zero-coupon rates derived from Government bond prices as suggested in the researched finance literature. In contrast, we apply the model on end-of-month swap rates with tenors $\tau_i \in \tau = \{\frac{1}{4}, \frac{1}{2}, 1, 2, 5, 10, 20, 30\}$ in years. The data sample of the swap rates starts in January 1999 and ends in August 2017. Figure 4.2 plots the evolution of the term-structure of the swap rates and the average swap rates curve over the time period. Inspection of the left plot reveals that the term-structure of the swap rates assumes a variety of shapes over time, including mainly steep or flat upwarded curves and rarely humped or inverted curves. The right plot illustrates that the average swap rates curve has an increasing and concave shape. The swap rates data has been made available for this research by the Austrian Treasury.

5. In-sample fit

In this section we examine the extended Nelson-Siegel and Svensson models in terms of their ability to reproduce the dynamics of the term-structure of interest rates within our data. In order the compare the goodness of the in-sample fit we define several measures that are commonly used in finance literature. Moreover, we investigate the dynamics of the estimated state variables. We expect to identify the characterisations of the individual state factors introduced in Diebold and Li (2006).
Moreover, Diebold et al. (2006) and Bolder and Deeley (2011) use diagnostic tools to analyse the dynamic interactions among the term-structure of interest rates and the macro economy. We follow this approach and apply the impulse response function in the VAR specifications of the models.

5.1. Overall in-sample fit and dynamics in the term-structure

In the examination of the overall in-sample fit, we compare descriptive statistics applied on the residuals between the actual swap rates sw_t and the fitted swap rates $\tilde{sw}_t$ obtained by our implemented models. In order to measure the ability of the models to fit the term-structure we use the root-mean-square error calculated over the term-structure at a specific point in time t,

$$RMSE_t^{curve} = \sqrt{\frac{\sum_{\tau_i \in \tau}(\tilde{sw}_t(\tau_i) - sw_t(\tau_i))^2}{\#\tau}}. \tag{5.1}$$

Descriptive statistics of $RMSE_t^{curve}$ over time t summarise the overall fit of the models to the term-structure. Moreover, we include measures to investigate the in-sample fit of the models at specific tenors of the term-structure, namely, the root-mean-square error (RMSE) on the individual tenors over time,

$$RMSE_{\tau_i} = \sqrt{\frac{\sum_t(\tilde{sw}_t(\tau_i) - sw_t(\tau_i))^2}{\#T}}, \tag{5.2}$$

with $\#T$ as the number of time points in our data sample, and the mean of the residuals at the individual tenors over time,

$$Mean_{\tau_i} = \frac{1}{\#T}\sum_t(\tilde{sw}_t(\tau_i) - sw_t(\tau_i)). \tag{5.3}$$

Table 5.1 presents descriptive statistics of $RMSE_t^{curve}$. Comparing the values associated to the Nelson-Siegel respectively to the Svensson model it is obvious that the latter is

Table 5.1.: **In-sample fit over term-structure curve:** We present the goodness of in-sample fit in terms of the overall fit to the term-structure curve. The table shows descriptive statistics of the root-mean-square error over the term-structure. The values are in percent.

Nelson-Siegel	Mean	Median	Std. Dev.	Min.	Max.
2-step fix	0.0752	0.0682	0.0386	0.0274	0.2730
2-step var	0.0640	0.0568	0.0324	0.0194	0.2055
1-step	0.0910	0.0825	0.0433	0.0262	0.2939
1-2-step	0.0734	0.0630	0.0399	0.0195	0.2668
Svensson	Mean	Median	Std. Dev.	Min.	Max.
2-step fix λ	0.0548	0.0454	0.0328	0.0147	0.1944
2-step var	0.0469	0.0379	0.0324	0.0095	0.1801
1-step	0.0836	0.0665	0.0556	0.0179	0.3375
1-2-step	0.0571	0.0457	0.0387	0.0116	0.2315

marginal superior over all descriptive statistics of $RMSE_t^{curve}$. The figures on the individual model specifications reveal that the best in-sample fit is achieved by the `2-step var` method in both models. These observations are consistent with the results of de Pooter (2007) indicating that more flexible model specifications achieve a better goodness of the in-sample fit. However, the improvements by the more flexible estimation methodology are not significant and the `2-step fix` method is overall competitive. Interestingly, the `1-step` method is outperformed by the other estimation approaches in terms of overall fit of the term-structure. Recap that the `1-2-step` approach uses the estimates of the decay parameters obtained by the one-step estimation. Therefore, the results indicate that the smoothed state variables differ from the state variables obtained by the `1-2-step` estimation approach. This might reflect the fact that the `1-step` method uses information from the observed interest rates within the Kalman filter as outlined by Diebold et al. (2006).

Table 5.2 presents the root-mean-square error at individual tenors $RMSE_{\tau_i}$, allowing a more detailed analysis of the model's in-sample fit per individual tenor. It reveals that the Svensson model slightly outperforms the Nelson-Siegel model on the majority of the tenors. Moreover, one can see that the `1-step` method is competitive or even superior to the other model specifications at most of the maturities, especially in the Svensson model set-up. But, on the contrary, it is inferior on the 2 and 30 years tenors which most likely cause the weaker overall in-sample fit observed in Table 5.1.

Until now, our analysis has not provided any insights if the models underestimate or overestimate the swap rates observed in our data. Therefore, we investigate the mean of the residuals per tenor, $Mean_{\tau_i}$. Positive values indicate that the swap rates are overestimated at tenor τ_i and vice versa.
Figure 5.1 presents the measure over the tenors for the Nelson-Siegel and Svensson model estimated by the different estimation methodologies. It reveals that among the two-step estimated model specifications there are no significant differences in regard to the sign and magnitude of the measure $Mean_{\tau_i}$ over the tenors. Starting with the underestimation of the short term tenors 3M and 6M by the models, the sign of the $Mean_{\tau_i}$ measure changes over the maturity. In summary, the models underestimate the swap rate at the tenors 3M, 6M, 2Y, 10Y and 20Y, and overestimate the swap rates at the tenors 1Y, 5Y and

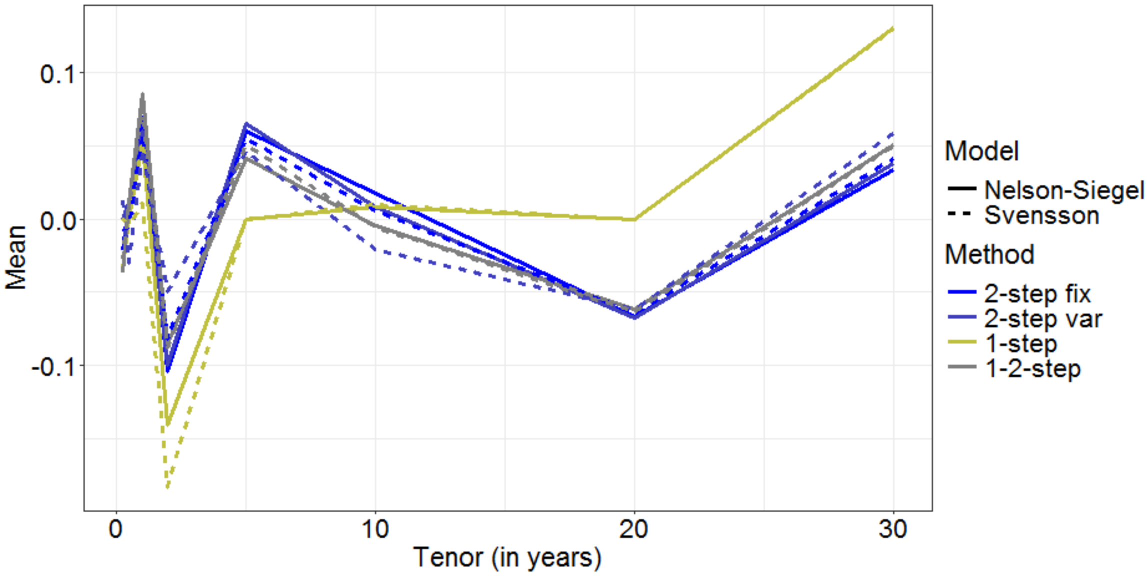
Table 5.2.: **In-sample fit over tenors:** We present the goodness of in-sample fit in terms of the fit over the tenors. The table shows the root-mean-square error per tenor τ_i. The values are in percent.

	Nelson-Siegel				**Svensson**			
	2-step		1-step		2-step		1-step	
Tenor	fix	var	1-step	2-step	fix	var	1-step	2-step
3 months	0.0722	0.0420	0.0949	0.0828	0.0165	0.0226	0.0031	0.0366
6 months	0.0204	0.0292	0.0000	0.0194	0.0247	0.0391	0.0140	0.0175
1 year	0.1264	0.0830	0.1179	0.1354	0.0706	0.0629	0.0223	0.0965
2 years	0.1176	0.1142	0.1739	0.1065	0.1007	0.0686	0.2315	0.1117
5 years	0.0780	0.0821	0.0000	0.0624	0.0710	0.0605	0.0000	0.0651
10 years	0.0630	0.0387	0.0439	0.0588	0.0430	0.0346	0.0374	0.0276
20 years	0.0742	0.0791	0.0000	0.0710	0.0758	0.0729	0.0002	0.0752
30 years	0.0779	0.0633	0.1617	0.0806	0.0632	0.0712	0.1577	0.0624

30Y. Moreover, the plot strengthens the observations on the `1-step` estimation approach. While, both models are inferior at the 2 years and 30 years tenors, they outperform the two-step estimation specifications at the short tenors and from tenors 5 to 20 years.

In the following we analyse the ability of the models to capture the dynamics and various shapes of the term-structure within our data in more detail. Therefore, figure 5.2 presents the evolution of $RMSE_t^{curve}$ over time t associated to the Nelson-Siegel and Svensson specifications. At first the plot strengthens the observations that the more sophisticated Svensson model outperforms the Nelson-Siegel model. Over a large part of the data the Svensson model is clearly superior to the Nelson-Siegel model. Among the different estimation methodologies the `2-step var` approach achieves not surprisingly over the entire data the best in-sample fit over the term-structure. We will observe in a subsequent analysis that the superiority in the in-sample fit is connected with less robust dynamics of the state variables.
A detailed analysis on the evolution of $RMSE_t$ over time resulted from the `1-step` and `1-2-step` methods brings interesting insights. At first it strengthens the observations from the tables that the `1-step` approach is outperformed by the other approaches in terms of the goodness of in-sample fit. Only at the end of our data it gets competitive. Moreover, in both models the `1-2-step` method gets superior at the latest 5-years of our data only slightly beaten by the `2-step var` approach. The results of the `1-2-step` is therefore interesting as it improves the in-sample fit of the two-step estimation approach with pre-specified decay parameters as suggested by Diebold and Li (2006). This indicates that the information on the decay parameters obtained by the one-step estimation approach has a beneficial impact on the models. Going into the detail of the estimation results of the `1-step` estimation approach, the decay parameter of the Nelson-Siegel model is estimated by $\lambda = 0.4803$. The linked curvature factor loading achieves its maximum at the 3.7 years tenor. In the Svensson model the decay parameters are estimated by the `1-step` method as follows: $\lambda^1 = 0.9953$ and $\lambda^2 = 0.4819$. The λ-values can be interpreted in the way that the curvature factor loading f_2 achieve its maximum at 1.8 years tenor, whereas the curvature factor loading f_3 achieve its maximum at 3.7 years tenor. The differences of the

Figure 5.1.: In-sample fit over tenors

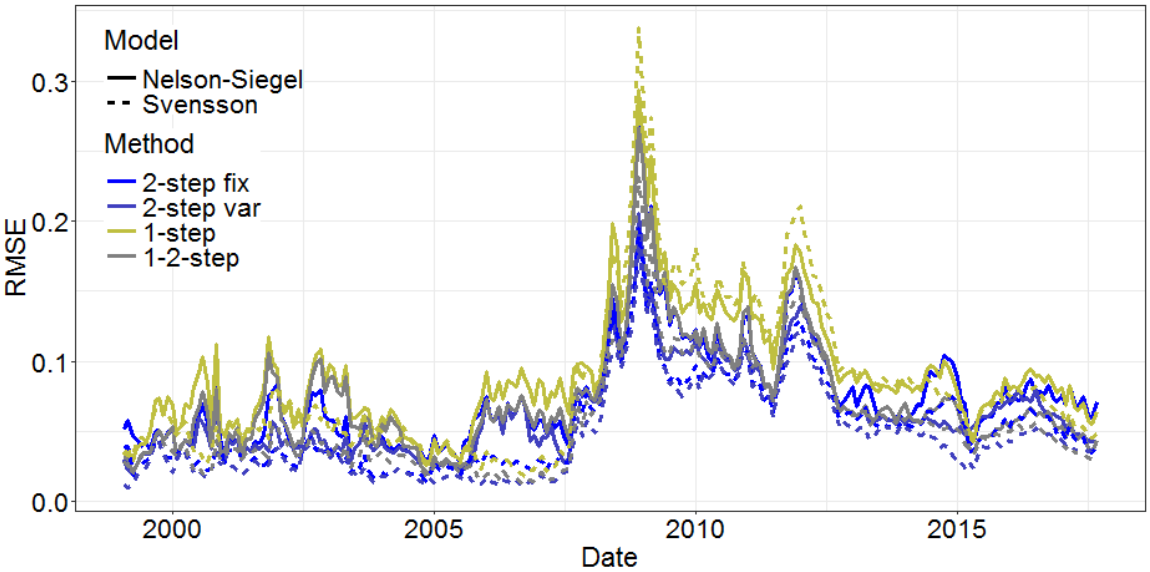


estimates of the decay parameters to the pre-specified values, as well as the observations on the in-sample fit analysis indicate that the values of the decay parameters may contain beneficial information on the shape of the term-structure of the interest rates.

Coming back to the bigger picture on the in-sample fit of the term-structure, the most general observation in Figure 5.2 linked to all models is the substantial volatility on the goodness of in-sample fit over time. All model specifications have particular difficulties to fit the term-structure in the time period from 2008 to 2012 within our data. Reviewing the swap rates data sample in the left plot of Figure 4.2 one can see that the term-structure becomes very flat in 2008 and 2009 followed by a massive decrease of the short tenor swap rates. These extraordinary dynamics may be linked on a very basic way to the global financial crises 2007-08 and the subsequent European debt crisis[1].
However, the dynamics in the term-structure during this period obviously cause difficulties to our implemented models.
In this regards, figure 5.3 presents various shapes of the term-structure and the average swap rate curve appearing in our data sample. It shows that the models are overall able to reproduce the average swap rate curve and the variety of shapes of the term-structure. The plot in the second row and column presents a typical shape of the term-structure appearing in the time period around 2009. Consistent with the results in Diebold and Li (2006) it reveals that the models have difficulties fitting the term-structure when it is dispersed by multiple interior minima and maxima. However, overall the models perform a reasonable job describing the dynamics of the term-structure in our data.

[1] See e.g. https://www.ecb.europa.eu/mopo/decisions/html/index.en.html, https://en.wikipedia.org/wiki/Financial_crisis_of_2007-08 and https://en.wikipedia.org/wiki/European_debt_crisis.

Figure 5.2.: In-sample fit over the term-structure over time



Recapitulating the examination of the models in terms of the in-sample fit, we can conclude that the more sophisticated Svensson model slightly outperforms the Nelson-Siegel model. The Svensson model is superior at the majority of the data sample including time periods with dispersed term-structure curve. Introducing more flexible estimation methodologies allowing the decay parameters to vary over time has improved the in-sample fit of the models. However, over the whole data sample the improvement of the in-sample fit is only marginal. Therefore, considering the in-sample fit performance of the models we can outline that the more straightforward two-step estimation approach with fixed decay parameters is competitive to the complex and numerical expensive non-linear estimation techniques.
The results of the one-step estimation approach are somehow two folded. On the one hand it is significantly superior over several tenors considering the root-mean-square error. On the other hand it is inferior at the tenors of 2 years and 30 years. However, we have identified that the estimation of the decay parameters by the one-step estimation approach has a beneficial impact on the models. In particular, a combined approach of the one-step and two-step estimation methods does a reasonable job in terms of in-sample fit. Considering also the lack of theoretical background on the pre-specification of decay parameters in the two-step estimation approach suggested by Diebold and Li (2006), the one-step estimation approaches are reasonable methods within the estimation methodologies.

5.2. Dynamics of the state variables

In this section we analyse the dynamics of the estimated state variables with the objective to identify their characterisations introduced in Diebold and Li (2006), and to observe differences in the dynamics resulting from the individual estimation methodologies.

Figure 5.3.: In-sample fit on the average swap rate curve and selected swap rate curves

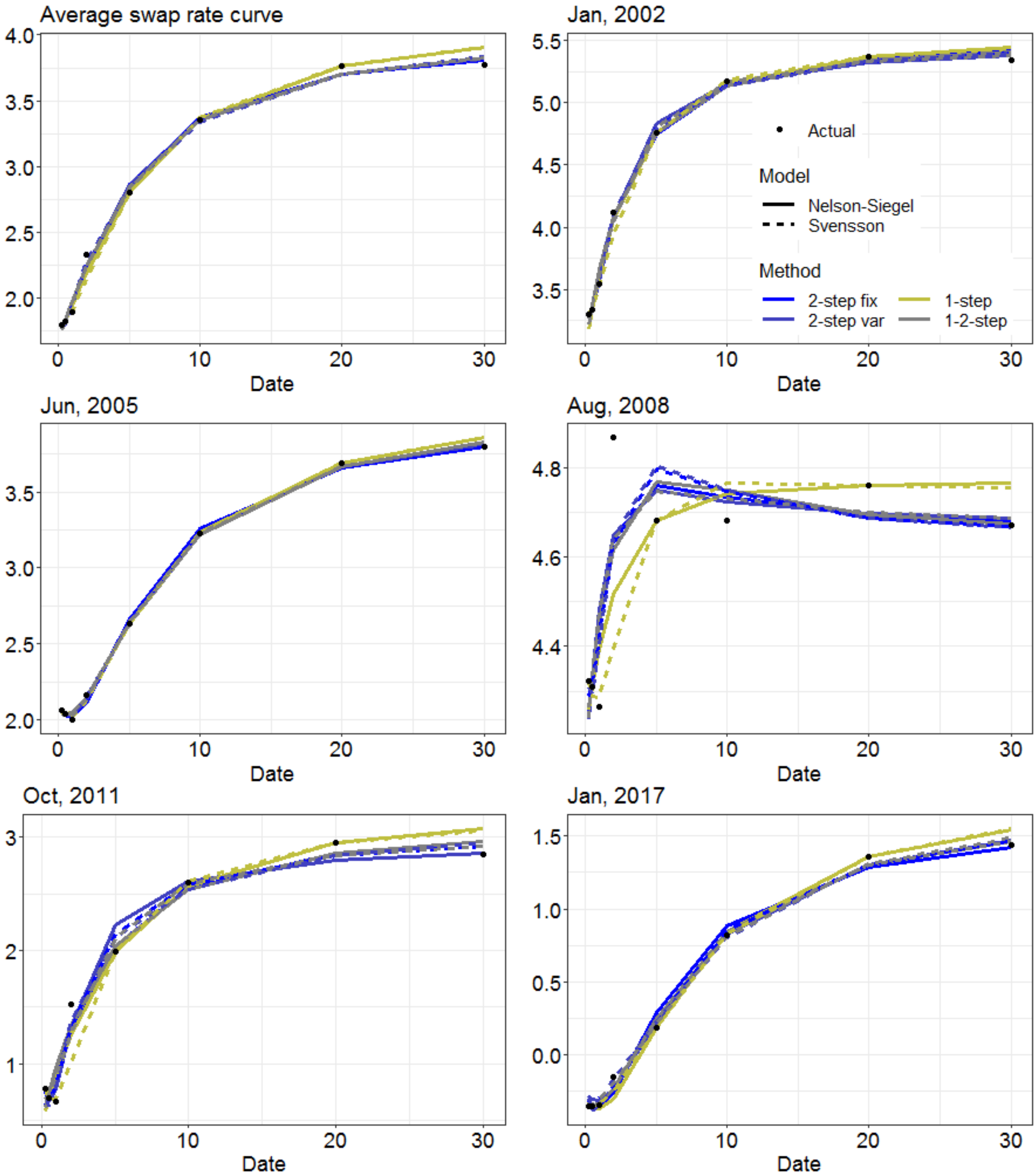
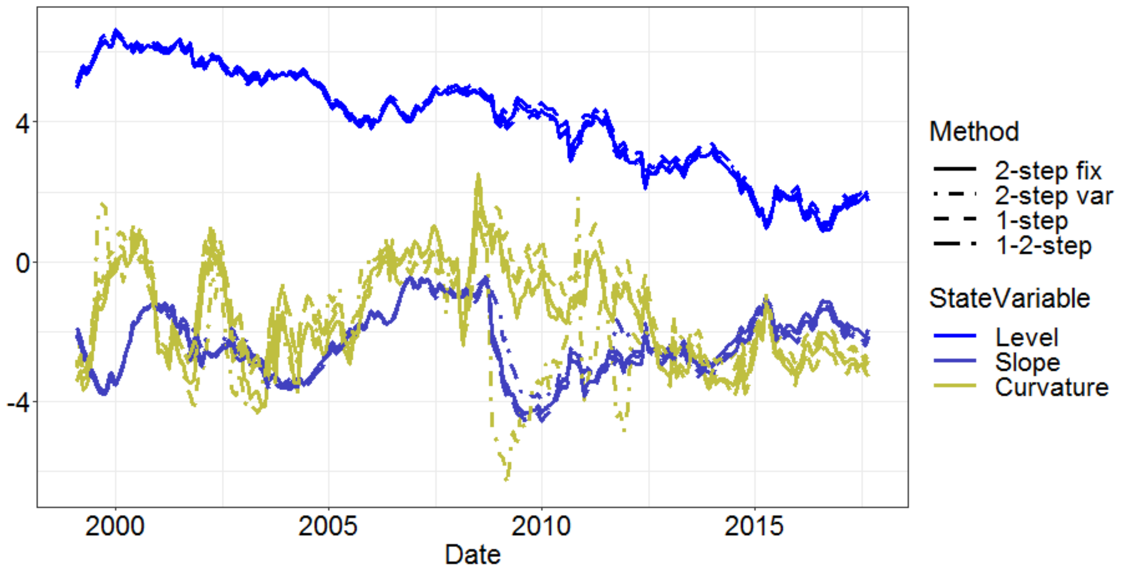


Figure 5.4.: Nelson-Siegel model - dynamics of the state variables



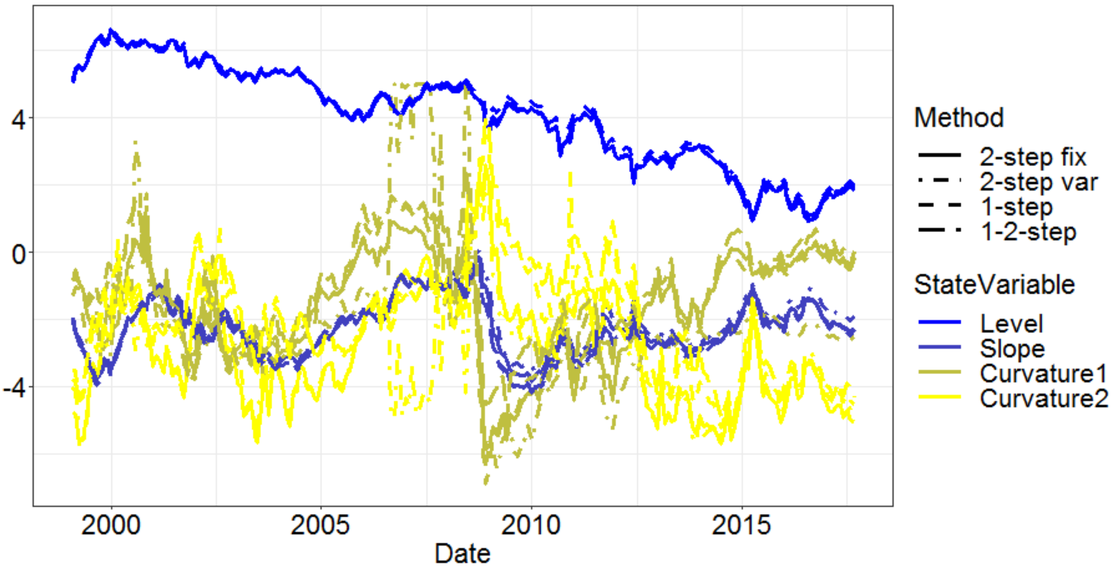
Recap that in the two-step estimation approaches the state variables are obtained using, either ordinary least squares after pre-specification of the decay parameters, or Differential evolution estimating the decay parameters simultaneously. On the other hand, the Kalman filter in the one-step estimation approach provides smoothed state variables based on the information of the observed swap rates in our data.

Figure 5.4 presents the dynamics of the state variables of the Nelson-Siegel model obtained by the different estimation methodologies. The level factors are obviously most persistent and identical to each other. In comparison to the dynamics of the swap rates in the left plot of figure 4.2 one can see that the decrease of the level factors is similar to the declining of the general level of the swap rates. This observation is consistent with Diebold and Li (2006) that the parallel shifts in the term-structure are reflected in the dynamics of the level factors.

The slope factors seem also to be quite robust over the entire data with few exceptions in the time period from 2009 to 2012. The interpretation of the values associated to the slope factors can be done reviewing the dynamics of the swap rates in Figure 4.2 as well. In particular, flat swap rate curves imply low negative slope values s_t, whereas high negative slope values occur during time periods with steep swap rate curves. Again, this is consistent with the results in Diebold and Li (2006) that the slope factor describes the steepening or flattening of the term-structure.

On the other hand, the dynamics of the curvature factors are exposed to substantial volatility over time. In particular, the numerical instability in the non-linear model specification of the Nelson-Siegel model is evident observing the dynamics of the curvature factors obtained by the two-step estimation approach with variable decay parameters. The dynamics are extraordinary, especially in the time period from 2009 to 2012. Overall, it is hardly possible to observe the characterisation of the curvature factor defined in Diebold and Li

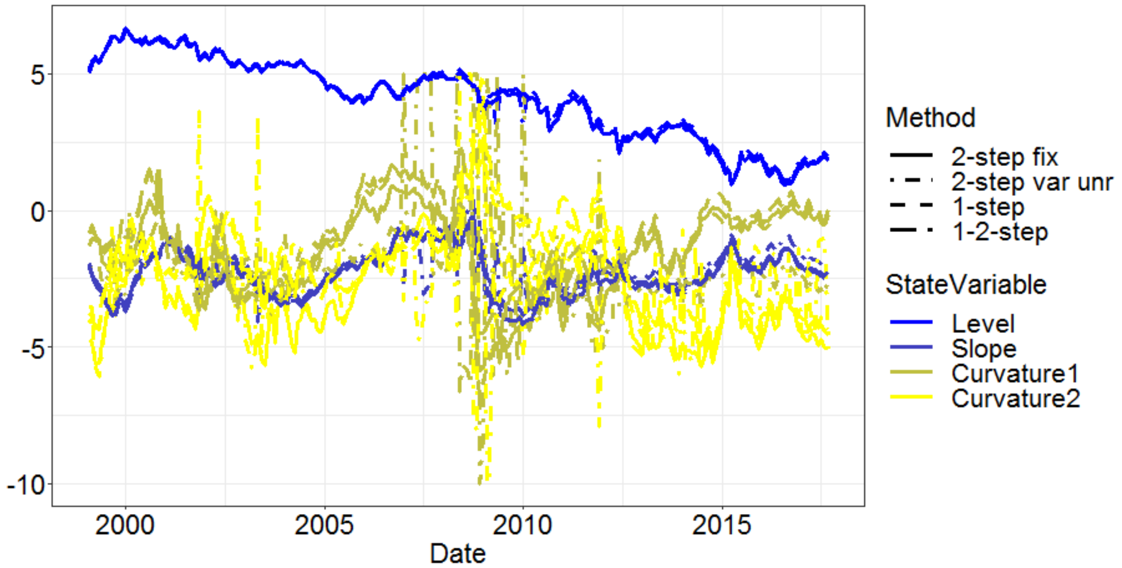
Figure 5.5.: Svensson model - dynamics of the state variables



(2006) by comparison to the dynamics of the swap rate curves.
To strengthen the assertions on the characteristics of the state variables in the Nelson-Siegel model, Diebold and Li (2006) define empirical versions of the level $\tilde{l}_t$, slope $\tilde{s}_t$ and curvature $\tilde{c}_t$ of a term-structure curve. The empirical versions are derived from the rates of the term-structure at different tenors, the empirical level is defined as the 10-year interest rate. Moreover, the empirical slope is defined as the 10-years minus the 3-months interest rates, and the empirical curvature as twice the 2-year interest rates minus the sum of the 3-month and 10-year interest rates. Calculating the correlation between the estimated Nelson-Siegel parameters and the empirical versions we get high correlation values close to $\pm$ 1 consistent with the results from Diebold and Li (2006).

Firgue 5.5 presents the dynamics of the state variables of the Svensson model obtained by the different estimation methodologies. The insights drawn from this figure are comparable to the Nelson-Siegel model.
Therefore, we can conclude for both models that the level and slope factors are robust and mostly consistent among the different estimation methods.
On the other hand, the dynamics of the curvature factors are exposed to significant volatility and it is hardly possible to identify their characterisation by analysing the dynamics of the swap rates. Moreover, the less robust dynamics of the curvature factors obtained by the `2-step var` approach indicate the numerical difficulties caused by the non-linear specifications of the models. Nevertheless, in order to present the positive impact of the restrictions on the λ-parametrization introduced in section 3.2.2, Figure 5.6 presents the dynamics of the state variables in the Svensson model obtained by the two-step estimation approach with unrestricted decay parameters. Obviously the volatility and magnitudes in the state variables are substantially greater, especially in the curvature factors. But

Figure 5.6.: Svensson model - dynamics of the state variables including two-step estimation approach with unrestricted decay parameters

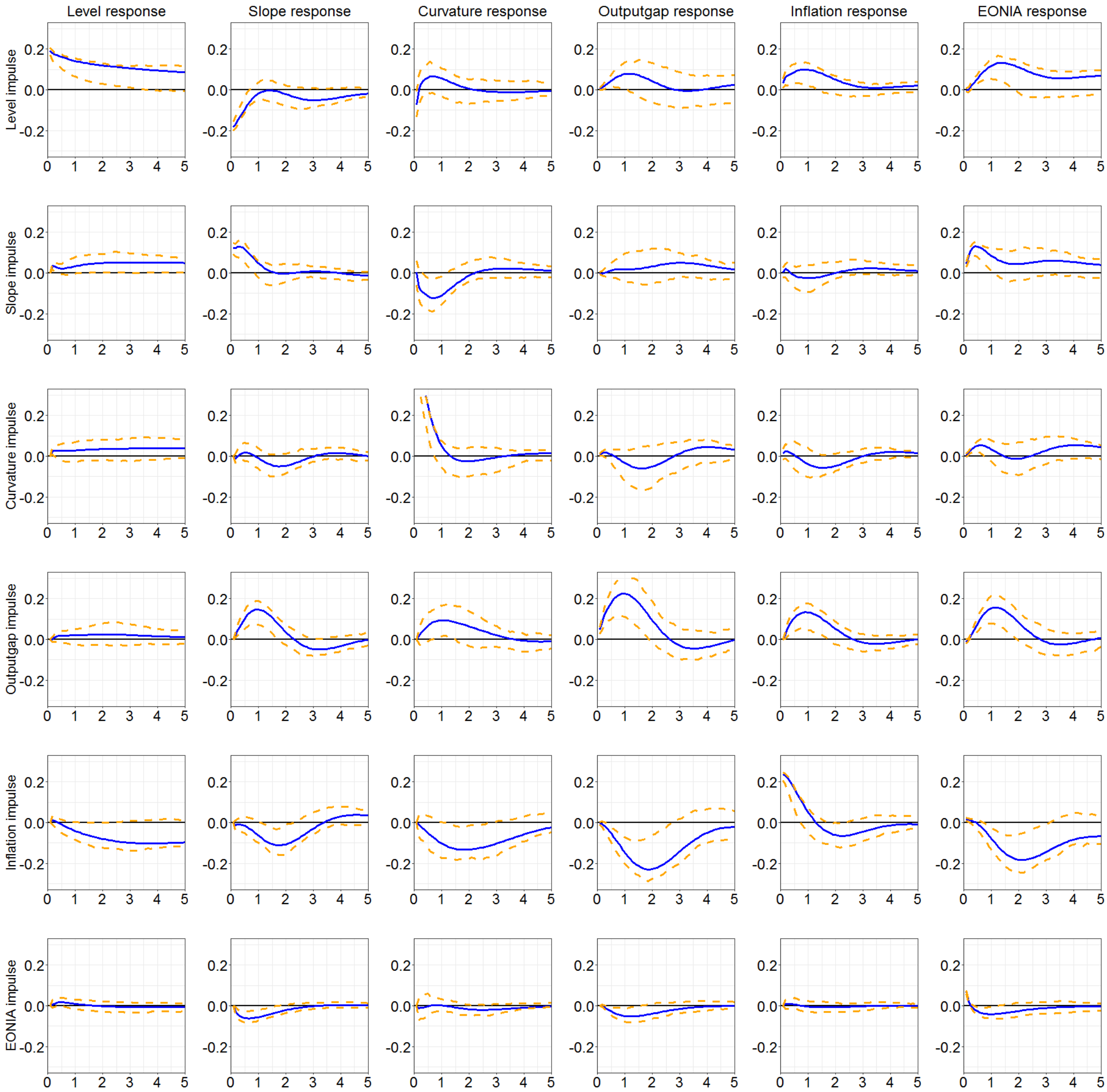


also the dynamics of the slope factor face extreme values multiple times. Therefore, we can conclude the beneficial impact of the imposed restrictions on the decay parameters suggested by de Pooter (2007) and Gilli et al. (2010).

5.3. Dynamic interactions among state variables and macroeconomic factors

One of the main motives for the incorporation of macroeconomic variables into a term-structure modelling framework is to analyse the dynamic interactions among the term-structure of interest rates and the macro economy. In this regards, Diebold et al. (2006) and Bolder and Deeley (2011) use as a diagnostic tool the impulse response function. The impulse response function allows us to analyse the effects of standard shocks on a state variable to other variables. In order to perform this analyses we follow Bolder and Liu (2007) and enlarge the estimated time series of the state variables by the macroeconomic factors and apply a VAR model. In this section and subsequently in the out-of-sample forecast analysis we use the **R**-package **vars** accompanying article Pfaff (2008b) and the respective book Pfaff (2008a). This package delivers functions for estimating vector autoregressive models (VAR) and performing various diagnostics, such as the impulse response function and forecast error variance decomposition. In order to compare the results with Diebold et al. (2006) and Bolder and Deeley (2011) we focus on the Nelson-Siegel model.
Figure 5.7 presents the orthogonal impulse responses among the state variables and macroeconomic factors. Starting with the macroeconomic variables we analyse the bottom right-hand 3x3 matrix.

Figure 5.7.: Impulse response function



Overall, one can see that a positive shock on the output gap is followed by a positive response of inflation and the EONIA as overnight rate. While the output gap rate and overnight rate display negative responses to a standard shock on the inflation. Moreover, we observe only marginal impact on the output gap rate and inflation rate after a change in the overnight rate. The results are overall very similar to Bolder and Deeley (2011) though the responding variables need longer to fall back to their initial value in our analysis. Bolder and Deeley (2011) have examined the dynamic interactions of the macroeconomic variables in their model framework to monetary policy macroeconomic models and find a general consistency. Therefore, we assume that the set-up of our model framework and the selected macroeconomic variables are reasonable.

Now, we are interested on the dynamic interactions between the state variables of the Nelson-Siegel model and the macro economy. The 3x3 matrix in the bottom left-hand corner presents the responses of the state variables resulted from standard shocks on the macroeconomic factors. In this regards, remind the charaterisations of the state factors described in section 2 and examined in our data in previous section 5.2. The level factor displays only modest responses to shocks on output gap rate and overnight rate. On the other hand, the inflation rate has a long lasting negative effect on the level factor which might reflect the steady decrease on the overall level of the swap rates which is highly correlated with the level factor.

The slope factor shows a positive response to a shock on the output gap rate which reflects a flattening of the term-structure. The argumentation in Bolder and Deeley (2011) seems reasonable also for our results. In fact, the increase in the output gap leads to an increase in the overnight rate as well, as both are highly correlated with the short-term tenors of the term-structure this leads to an increase of the short term tenors and a flattening of the curve.

In this regards, the negative response of the slope factor to a shock on the inflation rate can be argued again similarly to Bolder and Deeley (2011). Changes in the inflation rate are followed by negative responses in the output gap and overnight rate which leads to a steepening of the term-structure. The negative response of the slope factor reflects the steepening of the term-structure.

On the other hand, the marginal negative response of the slope factor to a positive shock on the overnight rate is quite counterintuitive.

The response of the curvature factor to shocks on the macroeconomic variables is more difficult to interpret. In general, the curvature factor displays a positive response to a shock on the output gap and a negative response to a shock on the inflation rate. A shock on the overnight rate produces no substantial response on the curvature factor.

While Diebold et al. (2006) observe that the curvature factor shows only marginal responses to shocks on the macro variables, we have identified respective interactions to the macro economy similar to the results in Bolder and Deeley (2011).

The top right-hand 3x3 matrix displays the responses of the macro economy following standard shocks on the state variables of the Nelson-Siegel model. A shock to the level factor causes overall positive responses to the macroeconomic variables. Especially the overnight rate shows a persistent response which is consistent with the result of Diebold et al. (2006), while Bolder and Deeley (2011) has identified marginal impact of the level factor to the macro economy. Shocks on the slope factor have generally less impact on the macro econ-

omy except on the overnight rate equivalent with Bolder and Deeley (2011). A positive shock on the slope factor reflects a flattening of the term-structure curve. Therefore, the positive response of the overnight which leads to an increase of the short-term rates of the term-structure and a flattening of the curve is reasonable. The responses to a shock on the curvature factor allow no clear intuitive interpretation.
In the top left-hand 3x3 we examine the interactions among the state variables. The slope and curvature factors return back to their initial value in a short period after the shock on the level factor. The negative response of the slope factor indicates the initially steepening of the term-structure after a positive shock on the level factor.
A shock on the slope factor generates a marginal but persistent increase in the level factor and a negative response of the curvature factor. Finally, a curvature shock produces a marginal but persistent increase in the level factor and an oscillating response of the slope factor around the initial value with marginal magnitudes.

Overall we have find in the examination of the impulse response function reasonable dynamics among the macroeconomic factors consistent with Diebold et al. (2006) and Bolder and Deeley (2011). This indicates an appropriate selection and inclusion of macroeconomic information in our model framework. Moreover, the results indicate strong evidence of dynamic interactions among the macro economy and the state variables. Though not all observations in the dynamic interactions can be intuitively interpreted the impulse response function provides a useful tool to analyse the dynamics among the model components.

6. Out-of-sample forecasting

The main motive for the introduction of the extended Nelson-Siegel model by Diebold and Li (2006) and subsequent extensions in other finance literature has been the formalisation of a term-structure model that does a reasonable job in describing and forecasting the term-structure of interest rates.

In this section we focus on the latter and examine the introduced models and its specifications in terms of their ability to forecast the term-structure of the swap rates within our data. In finance literature, the out-of-sample forecasting capability of term-structure models has been assessed in several aspects. We will partially follow the forecast analysis of Bolder and Liu (2007) whose results have motivated this research to focus on the extended Nelson-Siegel model and close variations. Inter alia, they compare the joint-macro term-structure models in terms of their forecast performance of the entire term-structure of interest rates and on individual tenors. As general benchmark they compare the models to the random walk assumption which postulates that the interest rates are martingales, i.e. that the conditional expectation of future interest rates for all forecasting horizons is the current term-structure curve. As we have introduced the extended Nelson-Siegel model as base model we use the original model approach suggested by Diebold and Li (2006) as benchmark and compare it to the introduced alternative model variations.

In the following, we introduce the forecast procedures for the models. Subsequently we present the results of the analysis on the forecast ability of the models. Based on the insights we will introduce an approach motivated by Bolder and Liu (2007) in order to take into account different economic regimes occurring within our data.

6.1. Forecasting procedures

In our model framework the forecasting of the swap rates require forecasts of the state variables.

Following Bolder and Liu (2007) we model the state variables enlarged by the macroeconomic factors with VAR models. Therefore, we use the **R**-package **vars** supplementing the article Pfaff (2008b) and the book Pfaff (2008a). The package delivers several functions to model time series by vector autoregressive models (VAR) as well as diagnostic tools to analyse VAR specifications and to identify plausible lag selection of the models. We have identified that the forecast performance of the models is overall robust using plausible VAR specifications with different time lags. In particular, the main characterisations in terms of forecasting performances among the model specifications of the Nelson-Siegel and Svensson model remains the same using different lags in the VAR models, therefore we present here the results obtained by VAR models with two time lags.

In finance literature, the majority of the forecast analyses use initial estimation periods of 10 to 15 years. We follow this approach and start the out-of-sample forecasts in our data at January 2009, this means we have ten years and 120 observations to initially estimate the VAR models. The estimated VAR specifications are used to predict the state variables for the forecast horizons of 1, 3, 6, 12 and 24 months. Finally, the forecasts of the state variables are inserted in the model formulations (2.10) or (3.1) with the respective λ-values. After each forecast step, we add one month to the underlying data period for estimation and forecast the swap rates for the defined forecast horizons as described above. The progress is done iteratively until the end of the entire data is reached. In the following, certain specifics in the forecast procedures induced by the different estimation methodologies are described.

The forecast procedure for the model specifications with two-step estimation approaches is quite straightforward. We use the information of the obtained state variables until the point in time t and forecast their values with the VAR model for the forecast horizons. Subsequently we insert the forecasts of the state variables into equation (2.10) or (3.1) with the respective decay parameters. For the approach with fixed decay parameters these are the λ parametrisations defined in section 3.2.1.
For the approach with variable decay parameters de Pooter (2007) suggests out of alternative choices to use the median of the λ_t^i estimates known up to time t. This is explained by the observation that the median of the time-series of decay parameters estimates provide more stable results than the alternatives using the mean or the latest decay parameter estimate.
However, we have identified that this approach is not overall competitive in our model framework and therefore we have used the most recent estimates at time t of the decay parameters as well. As a consequence the decay parameters are updated at each iterative step forecasting the swap rates. Hereinafter, the approaches are referred to as `2-step var-m` using the median and `2-step var-l` using the latest decay parameter estimate.

In terms of the one-step estimation approaches the out-of-sample forecasting is more sophisticated. If one would straightforward use the smoothed state variables estimated with the one-step estimation approach one would take into account information of observed interest rates of the entire data sample actually not known at the time of the out-of-sample forecast. Therefore, one need to perform the one-step estimation approach on the sub-data sample up to time t, estimate a VAR model on the obtained smoothed state variables and proceed as normal to forecast the swap rates over the forecast horizon. In the `1-2-step` method we have to apply the VAR model on the dynamics of state variables which are derived by ordinary least squares given the λ-values estimated by the `1-step` estimation approach on the sub-sample. For both approaches the decay parameters for the calculation of the swap rate forecasts are the ones estimated by the one-step estimation. At each forecast step the values of decay parameters are iteratively updated.

6.2. Out-of-sample forecast results

In order to compare the forecast ability of the models we follow Bolder and Liu (2007) and examine the models in terms of their forecast performance of the term-structure and the interest rates at the individual tenors. The measures are already defined by equations (5.1)-(5.3) and we only need to insert the forecast errors, namely $e_{t+h,t}(\tau_i) = s\tilde{w}_t(\tau_i) - sw_t(\tau_i)$. Table 6.1 presents the out-of-sample forecast performance of the models in terms of the goodness of the overall term-structure forecasts. The model specifications outperforming the original extended Nelson-Siegel model `2-step fix` are highlighted. A variety of observations can be made from the table. At first, we note that no combination of model and estimation methodology extraordinarily outperforms the original dynamic Nelson-Siegel model as suggested by Diebold and Li (2006) and extended by macroeconomic factors. The Svensson model remains in the forecast analysis in general the marginal superior model in describing the term-structure of the swap rates. However, we find that the slight superiority of the Svensson model is not linked to a beneficial characterisation in the dynamics of the four state factors allowing better forecasts, but rather to the marginal better fit of the term-structure as identified in the in-sample fit analysis in section 5.1. Examining the various model specifications one can see that the results of the more flexible models with variable decay parameters are over the forecast horizons not competitive to the `2-step fix` method. Especially, the `2-step var-m` approach is not competitive and even produces inferior results in the Nelson-Siegel model at short forecast horizons. In the Svensson model the results of the flexible model specifications are more robust but overall not competitive to the other variations. Interestingly, both the `2-step var-m` and the `2-step var-l` specifications outperform the other methods at the 24-month forecast horizon.

The estimation methods based on the one-step approach using the Kalman filter are over all forecast horizons competitive to the original extended Nelson-Siegel model. The stable but not superior results of the `1-step` method indicate that the smoothed state variables obtained by the Kalman filter do not have substantial beneficial properties improving the forecast performance of the models.

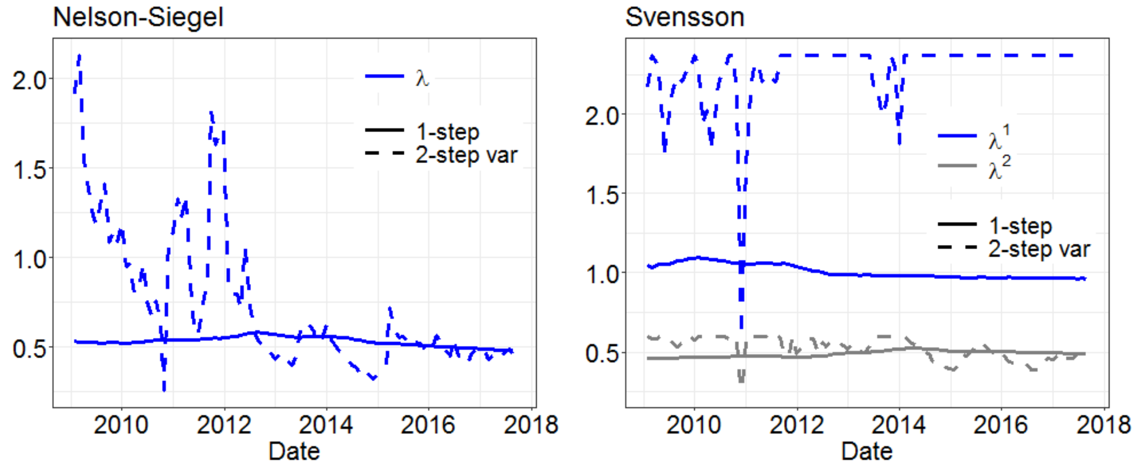
On the other hand, the results of the `1-2-step` method indicate that the combination of using estimates of the decay parameters and subsequently applying ordinary least squares provide reasonable results. While the `2-step fix` approach assumes the pre-specified values of the decay parameters to be constant over time, the `1-2-step` method use in each forecast step the latest information on the decay parameters slightly improving the forecast performance in the Nelson-Siegel model. This indicates that the estimates of the decay parameters may contain beneficial information on the current shape of the term-structure in the data.

In this regards, figure 6.1 presents the evolution of the decay parameters within the models estimated by the `1-step` and the `2-step var` estimation approaches. Both plots reveal that the one-step estimation approach gives varying but robust estimates of λ over the forecast period. While the λ values of the flexible two-step approach display a substantial volatility in the Nelson-Siegel model, especially in the time period from 2009 to 2012. In the Svensson model, the time-series of the λ-values estimated by the flexible 2-step approach are more stable with several extreme values. In general, the deviation to the pre-specified λ-parametrisations and the varying dynamics indicate that the decay parameters reflect certain characteristics of the term-structure curve as well. But in terms of the `2-step var` method it also displays the numerical instability of the decay parameters due to the non-linear specification of the models as described in section 3.2.2. de Pooter (2007) has suggested to use the median of the estimated λ-values based on observations of achieving

Table 6.1.: **Out-of-sample term-structure curve forecasts:** We present the goodness of forecast performance in terms of the overall fit to the term-structure curve. The table shows descriptive statistics of the root-mean-square error over the term-structure. The values are in percent.

	Nelson-Siegel					**Svensson**				
	Mean	Median	SD	Min	Max	Mean	Median	SD	Min	Max
One-month forecast										
2-step fix	0.1741	0.1558	0.0772	0.0690	0.4996	**0.1639**	**0.1446**	0.0778	0.0624	0.4921
2-step var-l	0.1763	0.1608	0.0888	0.0512	0.5095	0.1799	0.1632	0.0844	0.0626	0.5131
2-step var-m	0.2718	0.2399	0.1531	0.0788	0.7369	0.1904	0.1715	0.0973	0.0539	0.5218
1-step	0.1819	0.1629	0.0857	0.0557	0.5593	0.1837	0.1594	0.0903	0.0562	0.5599
1-2-step	**0.1709**	**0.1508**	0.0794	0.0620	0.4998	**0.1664**	**0.1419**	0.0793	0.0517	0.4944
Three-month forecast										
2-step fix	0.3321	0.3038	0.1469	0.0827	0.8883	**0.3260**	**0.2940**	0.1539	0.0799	0.8846
2-step var-l	0.3536	0.3422	0.1715	0.0786	0.9079	0.3653	0.3308	0.1660	0.0820	0.9416
2-step var-m	0.3895	0.3489	0.1899	0.1207	0.9155	0.3608	0.3536	0.1648	0.0816	0.8899
1-step	0.3325	0.3262	0.1496	0.0833	0.9112	0.3365	0.3213	0.1552	0.0773	0.9006
1-2-step	**0.3307**	**0.3033**	0.1481	0.0803	0.8912	**0.3277**	0.3057	0.1530	0.0874	0.9007
Six-month forecast										
2-step fix	0.5730	0.5354	0.3020	0.1128	1.6052	**0.5701**	**0.5338**	0.3148	0.0986	1.7051
2-step var-l	0.6229	0.5470	0.3586	0.1593	2.0562	0.6352	0.5457	0.3349	0.1151	1.9393
2-step var-m	0.6210	0.5826	0.3188	0.1050	1.6049	0.6227	0.5730	0.3279	0.1111	1.8933
1-step	**0.5653**	**0.5230**	0.3133	0.1116	1.5939	**0.5664**	0.5398	0.3252	0.0832	1.6822
1-2-step	**0.5725**	0.5380	0.3034	0.1148	1.6096	0.5743	0.5427	0.3069	0.1281	1.7187
Twelve-month forecast										
2-step fix	0.9853	0.7642	0.6650	0.1594	4.0735	**0.9712**	**0.7504**	0.6775	0.1671	4.0995
2-step var-l	1.0433	1.0204	0.6837	0.1707	4.1432	1.0641	0.8645	0.6735	0.1836	4.2936
2-step var-m	1.0227	0.8720	0.6507	0.1870	3.9519	1.0509	0.8428	0.6690	0.2174	4.2770
1-step	**0.9830**	0.7656	0.6784	0.1257	4.1357	**0.9701**	0.7595	0.6713	0.1633	3.8773
1-2-step	**0.9846**	**0.7611**	0.6661	0.1592	4.0768	**0.9745**	0.7568	0.6744	0.2079	4.2351
24-month forecast										
2-step fix	1.6266	1.4808	0.8016	0.3084	3.5452	1.6559	1.5623	0.8755	0.1935	3.4938
2-step var-l	**1.5616**	**1.4614**	0.7551	0.2776	3.6837	**1.5479**	1.4832	0.7186	0.3563	3.0830
2-step var-m	**1.5802**	**1.4157**	0.7207	0.4090	3.4416	**1.5472**	1.4873	0.7075	0.3620	3.0757
1-step	1.6451	1.5809	0.8195	0.3230	3.4853	1.7286	1.7765	0.8872	0.2164	3.3778
1-2-step	**1.6255**	**1.4785**	0.8032	0.3063	3.5394	**1.6151**	1.4979	0.8282	0.2436	3.4804

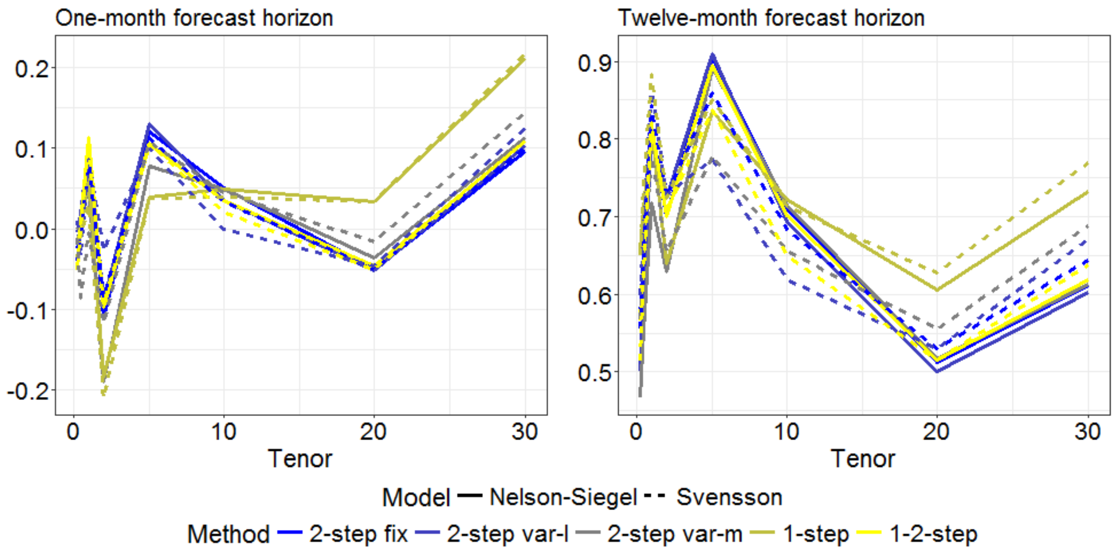
Figure 6.1.: Dynamics of the estimated decay parameters



more stable results in comparison to alternative choices. Considering the results of our out-of-sample forecast analysis we find that both approaches using either the median or the current value of the λ estimates are not sufficient to perform stable forecasts in our data. Moreover, an adequate approach to use the estimated decay parameters in the forecasting procedures seems dependent on the underlying data. Therefore, we clearly prefer the two-step estimation approach with fixed decay parameters and the one-step approaches.
With regards to the comparison of the `2-step fix` and `1-step` method, remind that Diebold et al. (2006) prefer the latter as it uses information from the observed interest rates for the estimation of the parameters and the smoothing of the state variables. On the other hand, Bolder adhere to the two step estimation approach in Bolder (2006) and Bolder and Liu (2007). He states in Bolder and Liu (2007) that they have experimented with the Kalman filter approach but have obtained superior results with the two-step estimation method with respect to their purpose. Based on our results we find similar to Bolder and Liu (2007) that the two-step estimation approach is more stable over all forecast horizons and therefore preferable for forecast activities. Moreover, the results indicate that the pre-specified λ values, in particular for the Nelson-Siegel model suggested by Diebold and Li (2006), already provide reasonable parametrisation of the decay parameters. On the other hand we have identified beneficial information in estimated decay parameters slightly improving the forecast ability of the models by combining the one-step and the two-step estimation approach. Therefore, we find it plausible to apply the one-step estimation approach in order to obtain information on reasonable values for the decay parameters.

Overall, the most general insight in Table 6.1 is that the forecast performance of all models substantially deteriorates increasing the forecast horizons. In this regards, Figure 6.2 presents the mean on the forecast errors at the individual tenor for the forecast horizons of one and twelve months. The left plot indicates that the general characterisations of the models in terms of term-structure fit remain unchanged to the observation in the in-sample fit analysis. Nevertheless, increasing the forecast horizon to twelve months one can see that the models are substantially overestimating the swap rates over the forecast period.

Figure 6.2.: Out-of sample forecasts over tenor

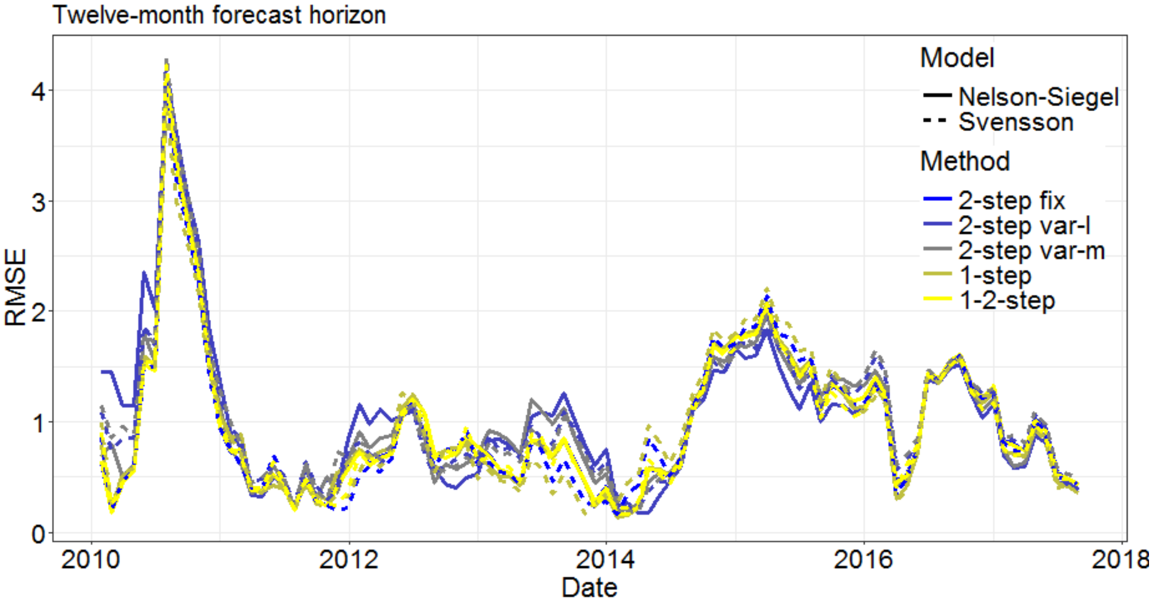


Reviewing the dynamics of the swap rates (figure 4.2) and the macro economy (figure 4.1), one can see that in the time period our out-of-sample forecast analysis has started a drastically collapse of the swap rates and the macro economy occurred. In addition, the swap rates have not returned to their historical average level but have even decreased furthermore over the out-of-sample forecasting period. Figure 6.3 displays the evolution of the root-mean-square error of the model forecasts of the term-structure at a point in time for a twelve-month forecast horizon. It reveals that all models have significant difficulties to forecast the term-structure in the time period after the substantial decrease in the interest rates in 2009, and again in the time period around 2015 after a further decrease of the swap rates. A detailed investigation of these time periods have revealed that the bad forecast performance of the models is related to a substantial overestimating of the swap rates. This observation is most likely explained by the inherent mean-reversion characteristics of the VAR models which drive the state variables and consequently the swap rates to return to their long term mean.

Bolder and Liu (2007) have made the same observations in their out-of-sample forecast analysis. They have identified as well that the models estimated on historical data have difficulties to model extraordinary decreases in interest rates and consequently overestimate interest rates over a longer time period.

They relate this problem to changes in the interest rates linked to different economic regimes in their data and that the constant-parameter assumption in the VAR specifications is not ideal. In particular, they outline that the estimation of a term-structure model using the entire data over all inherent economic regimes and the subsequent forecast simulation for risk management purposes would be hardly reasonable. The forecast simulation would not be an appropriate forward-looking description of the current economic regime occurring in the latest period of the data but a rather weighted average of different inherent economic regimes. In order to approach this problem they introduce time-varying parameters in the

Figure 6.3.: Out-of sample forecasts over time - twelve month forecast horizon



VAR specifications of the state variable dynamics. In particular, they impose exogenously different economic regimes identified in their data and link the intercept vectors of the VAR specifications to these regimes[1]. Therefore, depending on the economic regime within a time period of the data sample the intercept vectors of the VAR models change. As a result, Bolder and Liu (2007) have observed substantial improvements in the out-of-sample forecast ability of the models permitting certain parameters of the VAR specification to vary-over-time.

In the following we apply a simplified approach motivated by the suggestion of Bolder and Liu (2007). Recapping the results of the in-sample fit and out-of-sample forecast analysis so far, we have identified in the swap rates data the time period from 2008-2012 where all models have difficulties to describe the occurring term-structure curves of the swap rates. Moreover, after a massive decrease of the short-term rates the overall interest rates level further decreases and does not return to its long term value. We have linked the extraordinary dynamics in a straightforward manner to the global financial crisis 2007-08 and the European sovereign debt crisis. ECB has responded to the crises by introducing several non-standard monetary policy measures[2]. The dynamics of the swap rates indicates that the ECB's monetary policy decisions induce a very low interest rates environment not returning to its long term average level. Within the decisions of non-standard policy measures, a main key event has been the "whatever it takes" speech by former ECB President Mario Draghi, who has stated at the Global Investment Conference in London 26 July 2020: "Within our mandate, the ECB is ready to do whatever it takes

[1] The economic regimes in the Canadian macroeconomic factors imposed by Bolder and Liu (2007) have been identified in Demers (2003).

[2] https://www.ecb.europa.eu/mopo/decisions/html/index.en.html

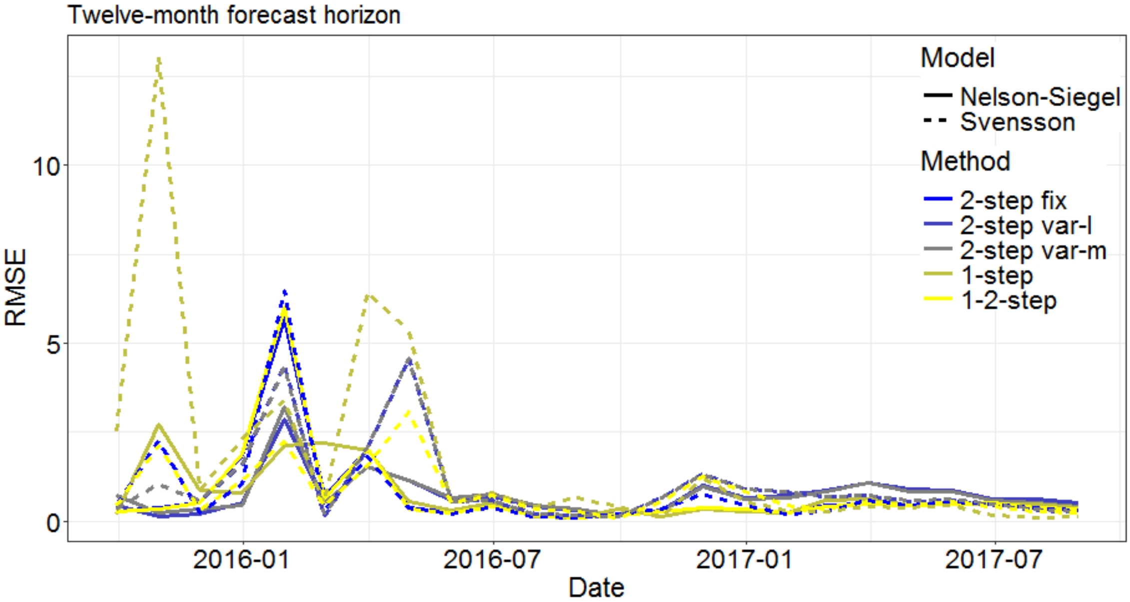
Table 6.2.: **Out-of-sample term-structure curve forecasts (2nd regime):** We present the goodness of forecast performance in terms of the overall fit to the term-structure curve. The table shows descriptive statistics of the root-mean-square error over the term-structure. The values are in percent.

	Nelson-Siegel					**Svensson**				
	Mean	Median	SD	Min	Max	Mean	Median	SD	Min	Max
					One-month forecast					
2-step fix	0.1546	0.1542	0.0689	0.0717	0.3354	0.1551	**0.1349**	0.0899	0.0548	0.4526
2-step var-l	**0.1406**	**0.1204**	0.0729	0.0519	0.3036	**0.1510**	**0.1287**	0.0892	0.0410	0.3604
2-step var-m	0.1738	**0.1449**	0.0911	0.0797	0.3715	0.1832	0.1574	0.0838	0.0823	0.4075
1-step	0.1548	**0.1425**	0.0709	0.0583	0.3762	0.1723	**0.1491**	0.1038	0.0597	0.4338
1-2-step	**0.1468**	**0.1418**	0.0733	0.0533	0.3269	**0.1515**	**0.1340**	0.0927	0.0456	0.5195
					Three-month forecast					
2-step fix	0.2580	0.2196	0.1500	0.0697	0.7239	**0.2504**	0.2267	0.1338	0.0800	0.6999
2-step var-l	**0.2393**	0.2314	0.1262	0.0452	0.6243	**0.2362**	**0.2007**	0.1287	0.0707	0.5852
2-step var-m	**0.2447**	**0.1966**	0.1422	0.0868	0.6354	**0.2565**	**0.2142**	0.1086	0.1267	0.5761
1-step	0.2674	0.2238	0.1619	0.0761	0.7915	**0.2492**	0.2282	0.1526	0.0834	0.9070
1-2-step	**0.2530**	**0.2151**	0.1511	0.0633	0.7130	**0.2357**	0.2624	0.1151	0.0825	0.4526
					Six-month forecast					
2-step fix	0.4062	0.3070	0.3421	0.1335	1.7705	**0.3757**	**0.3012**	0.2546	0.1153	1.2737
2-step var-l	**0.3775**	0.3533	0.2495	0.0454	1.1237	0.4228	0.3419	0.3321	0.0801	1.4748
2-step var-m	**0.3780**	0.3124	0.2429	0.1246	1.1512	0.4158	0.3242	0.3284	0.0826	1.4470
1-step	0.4111	0.3361	0.2951	0.1337	1.1486	0.4805	0.3719	0.4706	0.1164	2.5351
1-2-step	**0.4019**	**0.2975**	0.3465	0.1256	1.7951	**0.3501**	**0.3009**	0.2082	0.1112	1.0077
					Twelve-month forecast					
2-step fix	0.7168	0.3850	1.1586	0.1118	5.7078	0.7841	**0.3780**	1.3180	0.1141	6.4846
2-step var-l	0.7366	0.6162	0.5656	0.1403	2.8601	0.9936	0.5991	1.1691	0.1218	4.5950
2-step var-m	0.7558	0.6387	0.6108	0.1816	3.1982	0.9829	0.5842	1.1795	0.1298	4.5961
1-step	0.7211	0.4775	0.7387	0.1163	2.7139	1.7166	0.4999	2.9233	0.0855	13.00048
1-2-step	0.7235	**0.3813**	1.2174	0.1051	6.0022	0.8003	0.4940	0.7479	0.1109	3.0773

to preserve the euro. And believe me, it will be enough."[3]
In order to investigate the effects of considering different economic regimes within our data we follow Bolder and Liu (2007) in a straightforward manner. In particular, we impose exogenously as economic regime the time period after the "whatever it takes" statement. We extend the out-of-sample forecast analysis on the sub-sample from August 2012 to the end of our data. We start with the forecasts at September 2014 using 25 observations to estimate the VAR specifications of the state variables. This is in comparison to the previous forecast analysis a very short time period to estimate the VAR models, however, we find it still interesting how the models perform. Table 6.2 presents the out-of-sample forecast performance of the models in terms of the goodness of the overall term-structure forecasts limited on the time period of the identified economic regime in our data. The 24-month forecast horizon has been neglected due to the short forecast period. Overall the table reveals that all models achieve a better forecast performance especially at the longer

[3] https://www.ecb.europa.eu/press/key/date/2012/html/sp120726.en.html

Figure 6.4.: Out-of sample forecasts over time - twelve month forecast horizon



forecast horizons. The main characterisations among the different model specifications in terms of the forecast ability remain. The `2-step fix` and `1-2-step` methods achieve the most stable forecast performance over all forecast horizons. Although the more flexible estimation methodology with variable decay parameters improve substantially and becomes competitive in the shorter forecast horizons. This is very likely associated to less volatility in the dynamics of the estimated decay parameters over the shorter forecast period displayed in figure 6.1.

Nevertheless, the maximum values in the twelve-month forecast horizon reveal extreme values. Figure 6.4 shows the evolution of the root mean square error of the model forecasts over the term-structure at a point in time for a twelve-month forecast horizon in the identified economic regime. It reveals in the first phase of the forecast period inferior forecast performances of all models which is mostly likely related to the short estimation period of the VAR specifications. However, this phase is followed by a robust and good forecast performance of almost all models.

7. Conclusion

The research has started with a review of finance literature on joint-macro models that describe the dynamics of the macro economy and the term-structure of interest rates. With the objective to identify a term-structure model that is reasonable for application in a debt-strategy and risk management model framework of a Government this has been a challenging task as the majority of the papers did not focus on this aspect. Fortunately, the Bank of Canada has published numerous working papers investigating term-structure models and their usage for debt strategy and risk management problems of a Government. Leveraging on the results from Bolder (2006), Bolder and Liu (2007) and Bolder and Deeley (2011), we have focused on the extended Nelson-Siegel model suggested by Diebold and Li (2006).

The extended Nelson-Siegel model is a dynamic model describing the dynamics of the term-structure of interest rates by three latent state variables. Associated to their characteristics the state variables are known as level, slope and curvature factors.

The examination of the extended Nelson-Siegel model and further finance literature has indicated that the inclusion of further developments on the model might be rewarding. The introduction of the Svensson model has been motivated by the results of de Pooter (2007), who has achieved better in-sample fit and out-of-sample forecasts of the term-structure of interest rates by more sophisticated models.

The investigation on further estimation methodologies has been triggered by the lack of a theoretical foundation of the pre-specification of the λ-parametrisation in the Nelson-Siegel model as suggested by Diebold and Li (2006). In this regards, we have introduced three additional estimation methods. The most flexible approach uses Differential Evolution and permits the estimation of the state variables and decay parameters simultaneously at one point in time. We have described the non-linear specifications of the models causing the numerical difficulties of estimating robust state variables. Consequently, we have followed Gilli et al. (2010) and de Pooter (2007) by imposing restriction on the λ-values to approach the multicollinearity problem in the models and obtain robust state variables.

Moreover, we have included the developments of Diebold et al. (2006), who reformulate the extended Nelson-Siegel model into state-space representation and introduce an one-step estimation approach using the Kalman filter. The Kalman-filter allows the estimation of the model parameters, including the decay parameters, and the derivation of smoothed state variables in one-step considering the information in the observed interest rates. Therefore, Diebold et al. (2006) prefer the one-step estimation approach as the two-estimation approach lacks from omitting the information on the uncertainty fitting the observed interest rates associated in the first step.

Utilising the decay parameter estimates we apply the two-step estimation approach with the estimated λ-values and define a combined estimation approach by the one-step-two step estimation approach.

In order to incorporate macroeconomic factors in the model frameworks of the Nelson-Siegel and Svensson models we have followed Bolder and Liu (2007). The approach is straightforward by enlarging the times series of the estimated state variables level l_t, slope s_t and curvature c_t with selected macroeconomic variables. Following the finance literature we have selected as macroeconomic factors the annual inflation and output gap rate of the Economic and Monetary Union of the European Union (EMU). The monetary policy instrument in our model framework is set by the Euro Overnight Index Average (EONIA).

Consequently, the joint-macro models are examined in terms of their in-sample fit and out-of-sample forecast abilities.
In the in-sample fit analysis we have examined the ability of the models to describe the dynamics of the term-structure of the swap rates inherent in our data. We have observed that the more sophisticated Svensson model slightly outperforms the Nelson-Siegel model in terms of fitting the term-structure of the interest rates. Also the more flexible estimation methodology including variable decay parameters marginal improve the in-sample fit of the models. However, the more straightforward two-step estimation approach with fixed decay parameters is competitive. In particular, all model specifications perform a reasonable job describing the dynamics of the term-structure in our data. Although, the one-step estimation approach achieve less robust and stable goodness of in-sample fit due to inferior fits at the tenors of 2 and 30 years. However, we have identified that the estimation of the decay parameters by the one-step estimation approach has a beneficial impact on the in-sample fit of the models. In particular, a combined approach of the one-step and two-step estimation approach does a reasonable job in terms of in-sample fit. Considering also the lack of theoretical background on the pre-specification of decay parameters in the two-step estimation approach suggested by Diebold and Li (2006), the one-step estimation approaches are reasonable methods within the estimation methodologies.
Analysing the dynamics of the state variables we have observed that the level and slope factors are robust and mostly consistent among the different model specifications. Moreover, we have identified their characterisations as described by Diebold and Li (2006). On the other hand, we have observed significant volatility in the dynamics of the curvature factors and find it hardly possible to identify characterisation by analysing the dynamics of the swap rates. Moreover, the less robust dynamics of the curvature factors obtained by the `2-step var` approach indicate the numerical difficulties caused by the non-linear specifications of the models. Nevertheless, we have presented the positive impact of the restrictions on the λ-parametrisation suggested by de Pooter (2007) and Gilli et al. (2010).
In order to analyse the dynamic interactions among the term-structure of interest rates and the macro economy we have applied the impulse response function. We have find reasonable dynamics among the macroeconomic factors consistent with Diebold et al. (2006) and Bolder and Deeley (2011) indicating an appropriate selection and inclusion of macroeconomic information in our model framework. Moreover, we have identified substantial dynamic interactions among the macro economy and the state variables. Though not all observations in the dynamic interactions can be easily interpreted the impulse response function have provided useful insights on the dynamics among the model components.
Finally, we have examined the models in terms of their out-of-sample forecast ability. We have observed that no combination of model and estimation methodology extraordinar-

ily outperforms the original dynamic Nelson-Siegel model as suggested by Diebold and Li (2006). The Svensson model remains in the forecast analysis in general the marginal superior model in describing the term-structure of the swap rates. However, we find that the slight superiority of the Svensson model is not linked to a beneficial characterisation in the dynamics of the four state factors allowing better forecasts, but rather to the marginal better fit of the term-structure as identified in the in-sample fit analysis. In terms of the different estimation approaches we find that the more flexible methods with variable decay parameters are for both models not competitive due to less robust and stable forecast performance over the forecast horizons. The estimation methods based on the one-step approach using the Kalman filter are over all forecast horizons competitive to the original extended Nelson-Siegel model. The stable but not superior results indicate that the smoothed state variables obtained by the Kalman filter do not have substantial beneficial properties improving the forecast performance of the models.
On the other hand, the combined approach `1-2-step` using estimates of the decay parameters and subsequently applying ordinary least squares provide reasonable and partially superior results. Overall we find that the two-step estimation approach is preferable as providing more stable forecast results over all forecast horizons. This is consistent with Bolder and Liu (2007), who states that they have experimented with the Kalman filter approach but have obtained superior results with the two-step estimation method with respect to their purpose. Moreover, the pre-specification of the λ values, in particular for the Nelson-Siegel model suggested by Diebold and Li (2006), already provides good results in our analysis. Still, we find it reasonable to include the one-step estimation approach in the model framework as we have identified beneficial information in estimated decay parameters.

Overall, the research of the literature and our analyses indicate that the extended Nelson-Siegel model, introduced by Diebold and Li (2006) and applied by Bolder and Deeley (2011) in the Canadian debt-strategy model, is a reasonable starting point for jointly describing the dynamics of the term-structure of interest rate and the dynamics of macroeconomic factors. Holler et al. (2018) describe the application of the extended Nelson-Siegel model in a macro-financial framework for the Austrian perimeter. In particular, the model is used to describe and forecast the euro-area term-structure of interest rates and the Austria yield curve with the aim to enable risk management and debt-strategy analyses.

However, the research in our paper can be extended in numerous ways to further investigate on term-structure models and their application in a debt-strategy and risk management model framework of a Government. In particular, Bolder and Deeley (2011) outlines that not less than five stochastic models are used in the Canadian debt-strategy model, inter alia, to mitigate model risk. In this regards, Bolder and Romanyuk (2008) examine several combining techniques of term-structure forecasts calculated by different models finding that this averaging generally assists in mitigating model risk.

Moreover, the reformulation of the model in Diebold et al. (2006) has been a fundamental development that have implied further relevant evolvements of the extended Nelson-Siegel model. Even though Bolder and Deeley (2011) outline that there is reasonable amount of empirical evidence that empirical models outperform no-arbitrage models in terms of out-of-sample forecasting there are potential benefits also researching in this direction. In particular, Christensen et al. (2011) derive the class of arbitrage-free Nelson-Siegel models and show that the arbitrage free models can provide reasonable in-sample fit and out-of-sample forecasts as well.

Bibliography

Ang, A. and Piazessi, M. (2003). A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. *Journal of Monetary Economics, 50, 745-787.*

Björk, T. and Christensen, B. (1999). Interest Rate Dynamics and Consistent Forward Rate Curves. *Mathematical Finance , 9, 323-348.*

Bolder, D. J. (2006). Modelling Term-Structure Dynamics for Risk Management: A Practitioner's Perspective. *Bank of Canada Working Paper No. 2006-48.*

Bolder, D. J. and Deeley, S. (2011). The Canadian Debt-Strategy Model: An Overview of the Principal Elements. *Bank of Canada Discussion Paper No. 2011-3.*

Bolder, D. J. and Liu, S. (2007). Examining Simple Joint Macroeconomic and Term-Structure Models: A Practitioner's Perspective. *Bank of Canada Working Paper No. 2007-49.*

Bolder, D. J. and Romanyuk, Y. (2008). Combining Canadian Interest-Rate Forecasts. *Bank of Canada Working Paper No. 2008-34.*

Christensen, J., Diebold, F. X., and Rudebusch, G. D. (2011). The affine arbitrage-free class of Nelson-Siegel term structure models. *Journal of Econometrics*, 164(1):4–20.

de Pooter, M. (2007). Examining the Nelson-Siegel Class of Term Structure Models. *Tinbergen Institute Discussion Paper TI 2007-043/4.*

Demers, F. (2003). The Canadian Phillips Curve and Regime Switching. *Bank of Canada Working Paper No. 2003-32.*

Diebold, F. X. and Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics, 130, 337-364.*

Diebold, F. X., Rudebusch, G. D., and Aruoba, S. B. (2006). The macroeconomy and the yield curve: a dynamic latent factor approach. *Journal of Econometrics, 131, 309-338.*

Gilli, M., Große, S., and Schumann, E. (2010). Calibrating the Nelson-Siegel-Svensson model. *COMISEF WPS-031.*

Gilli, M., Maringer, D., and Schumann, E. (2019). *Numerical Methods and Optimization in Finance*. Elsevier/Academic Press, Waltham, MA, USA, second edition. ISBN 978-0128150658.

Holler, J., Nebenfuehr, G., and Radek, K. (2018). Forecasting the term-structure of euro area swap rates and Austrian yields based on a dynamic Nelson-Siegel approach. *Austrian Fiscal Advisory Council - Staff Papers*.

Litterman, R. and Scheinkman, J. (1991). Common Factors Affecting Bond Returns. *Journal of Fixed Income 1, 54–61*.

Luethi, D., Erb, P., and Otziger, S. (2018). *FKF: Fast Kalman Filter*. R package version 0.1.5.

Nelson, C. R. and Siegel, A. F. (1987). Parsimonious Modelling of Yield Curves. *The Journal of Business, Volume 60, Issue 4,473-489*.

Petris, G. (2010). An R Package for Dynamic Linear Models. *Journal of Statistical Software*, 36(12):1–16.

Petris, G., Petrone, S., and Campagnoli, P. (2009). *Dynamic Linear Models with R*. useR! Springer-Verlag, New York.

Pfaff, B. (2008a). *Analysis of Integrated and Cointegrated Time Series with R*. Springer, New York, second edition. ISBN 0-387-27960-1.

Pfaff, B. (2008b). VAR, SVAR and SVEC Models: Implementation Within R Package vars. *Journal of Statistical Software*, 27(4).

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Schumann, E. (2011–2019). *Numerical Methods and Optimization in Finance (NMOF) Manual. Package version 2.0-1*.

Svensson, L. E. O. (1992-1994). Estimating and Interpreting Forward Interest Rates: Sweden. *NBER Working Paper Series, 4871*.

Tusell, F. (2011). Kalman Filtering in R. *Journal of Statistical Software*, 39(2).

A. R-code

A.1. Additional R-packages

```r
## Additional Packages
# Graphical presentation of data analyses
library(ggplot2)
# Data transformation for graphical presentation
library(reshape2)
# Arrangements of plots within in one graphic
library(gridExtra)
# Time-indexed time series
library(xts)
# Colorisation of graphics
library(colorRamps)
# Combining plots in one graph with regards to legends
library(patchwork)

## Packages cited in the paper
# Differential Evolution for 2-step var method
library(NMOF)
# Estimating VAR models and applying related diagnostic tools
library(vars)
# Fast Kalman Filter for applying the Kalman filter, calculating the respective
    likelihood function
library(FKF)
# Applying the Kalman filter, calculating the respective likelihood function
library(dlm)
# Standard R-package providing maximum likelihood estimation
library(stats4)
```

A.2. Basic functions

The following code includes: (2.9), (2.10), (3.1), (5.1), (5.2), (5.3)

```r
## Factor loadings
# Arguments:
# - l ... lambda value (decay parameter)
# - tau ... tenor
slope_loading <- function(l,tau){
  (1 - exp(-l * tau))/(l * tau)
}
curvature_loading <- function(l,tau){
  (1 - exp(-l * tau))/(l * tau) - exp(-l * tau)
}

## Nelson-Siegel and Svensson model
# Arguments:
#  - tenor ... vector of tenors 0.25 years, 0.5 years, ..., 30 years
#  - param ... vector of parameters
#       - Nelson-Siegel = c(level, slope, curvature, lambda)
#       - Svensson = c(level, slope, curvature1, curvature2, lambda1, lambda2)
```

```
## Output: term-structure curve
nelson_siegel_model <- function(param,tenor){
  param[1:3] %*% rbind(rep(1,times=length(tenor)),
                       slope_loading(param[4],tenor),
                       curvature_loading(param[4],tenor))
}
svensson_model <- function(param, tenor){
  param[1:4] %*% rbind(rep(1,times=length(tenor)),
                       slope_loading(param[5],tenor),
                       curvature_loading(param[5],tenor),
                       curvature_loading(param[6],tenor))
}

## Measures to examine the in-sample fit and the out of sample forecast abilities
# RMSE (root-mean-square error) over term-structure curve
# Argument: x ... vector of residuals
RMSE_fun <- function(x){
  sqrt(sum((x^2))/length(x))
}

# descriptive statistics of the RMSE over term-structure curve
# Arguments:
#    # list_residuals ... list of residuals per model specifications
#    # str_type ... string of names of model specifications
#    # str_date_start/end ... time period on which the descriptive statistics are
    calculated
table_termstructure <- function(list_residuals,str_type, str_date_start = NULL,
    str_date_end = NULL){
  if ( !length(str_date_start ) ){
    str_date_start <- index(list_residuals[[1]])[1]
  }
  if ( !length(str_date_end ) ){
    str_date_end <- last(index(list_residuals[[1]]))
  }
  print(str_date_start)
  print(str_date_end)
  for ( i in str_type){
    # residuals
    res <- list_residuals[[i]][paste0(str_date_start,"/",str_date_end)]
    # RMSE over term-structure
    temp <- apply(res,1, function(x) RMSE_fun(x))
    # descriptive statistics
    if ( i == str_type[1]){
      df_residual <- cbind(mean(temp),median(temp),sd(temp),min(temp),max(temp))
    } else {
      df_residual <- rbind(df_residual,
                           cbind(mean(temp),median(temp),sd(temp),min(temp),
                               max(temp)))
    }
  }
  colnames(df_residual) <- c("Mean","Median","Std. Dev.","Min.","Max.")
  rownames(df_residual) <- str_type
  round(df_residual,4)
}
# RMSE and Mean per individual tenors
table_tenor <- function(list_residuals,str_type, str_date_start = NULL, str_date_end
     = NULL){
  if ( !length(str_date_start ) ){
    str_date_start <- index(list_residuals[[1]])[1]
  }
  if ( !length(str_date_end ) ){
    str_date_end <- last(index(list_residuals[[1]]))
  }
```

```
  print(str_date_start)
  print(str_date_end)
  residuals_statistics <- list()
  residuals_statistics[["Mean"]] <- residuals_statistics[["RMSE"]] <- NULL
  for ( k in str_type){
    res <- list_residuals[[k]][paste0(str_date_start,"/",str_date_end)]
    residuals_statistics[["Mean"]] <- cbind(residuals_statistics[["Mean"]],
                                            round(apply(res, 2, mean),4))
    residuals_statistics[["RMSE"]] <- cbind(residuals_statistics[["RMSE"]],
                                            round(apply(res, 2, function(x) RMSE_fun(x)),4))
  }
  colnames(residuals_statistics[["Mean"]]) <- colnames(residuals_statistics[["RMSE"]]) <- str_type
  residuals_statistics
}
```

A.3. Factor loadings

The following code includes: Figure 2.1, Figure 3.1

```
###########################
## Graphical presentation of the factor loadings
# set lambda-values
 lambdas <- c(0.6, 0.12) # NelsonSiegel or
# lambdas <- list(v1 = c(1.8, 0.3), v2 = c(0.6, 0.6)) # for Svensson
tau_graphic <- c(1/365,1/4,1/2,1,2,5,10,15,20,25,30)
# storage of plots
list_plots <- list()
for ( i in 1:2){
  # Nelson-Siegel
  data_temp <- data.frame(tau_graphic,
                          rep(1,length(tau_graphic)),
                          slope_loading(lambdas[i],tau_graphic),
                          curvature_loading(lambdas[i],tau_graphic))
  colnames(data_temp) <- c("Tenor","f0","f1","f2")
  # Svensson
  # data_temp <- data.frame(tau_graphic,
  #                          rep(1,length(tau_graphic)),
  #                          slope_loading(lambdas[[i]][1],tau_graphic),
  #                          curvature_loading(lambdas[[i]][1],tau_graphic),
  #                          curvature_loading(lambdas[[i]][2],tau_graphic))
  # colnames(data_temp) <- c("Tenor","f0","f1","f2","f3")

  data_temp <- melt(data_temp,"Tenor")
  list_plots[[i]] <- ggplot(data = data_temp, aes(x = Tenor,y = value, col = variable, linetype = variable))+
    geom_line(size=1.2)+
    labs(y = "",x="Tenor (in years)",
    # Nelson-Siegel or
         title=substitute(paste("Nelson-Siegel functions with ",lambda[t]," = ", value),list(value=lambdas[i])))+
    scale_color_manual(labels = c(expression(f[0],f[1],f[2])),values = rep("black",times=3))+
    scale_linetype_manual(labels = c(expression(f[0],f[1],f[2])),values = c(1,2,3))+
    # Svensson
    #    title=substitute(paste("Svensson functions: ",lambda[t]^1," = ",value1, " and ", lambda[t]^2," = ",value2),list(value1=lambdas[[i]][1],value2=lambdas[[i]][2])))+
    # scale_color_manual(labels = c(expression(f[0],f[1],f[2],f[3])),values = rep("black",times=4))+
```

```
    # scale_linetype_manual(labels = c(expression(f[0],f[1],f[2],f[3])),values = c
        (1,2,3,4))+
    theme_bw()+
    theme(legend.title=element_blank(),
          legend.key.width = unit(3,"line"),
          axis.title = element_text(size = 22,face="plain"),
          axis.text = element_text(size = 22,face="plain",color="black"),
          legend.text = element_text(size = 22,face="plain"),
          plot.title = element_text(size = 24,face="plain"),
          #legend.justification = c(0.8,0.8), # c(x,y) zwischen 0 und 1
          legend.position = c(0.8,0.7),
          plot.margin = unit(c(0,0.5,0,0),"cm")
    )
}
grid.arrange(arrangeGrob(list_plots[[1]]+theme(legend.position = "none"),list_plots
    [[2]],ncol=2))
```

A.4. Data presentation

The following code includes: Figure 4.1, Figure 4.2

```
###################
## Data presentation
## Swap rates data
# swap rates stored in xts object
head(xts_swap_rates, n = 4)
#               3M    6M    1Y    2Y    5Y   10Y  20Y  30Y
# 1999-01-31 2.995 2.925 2.860 3.015 3.446 4.024 4.58 4.75
# 1999-02-28 3.070 3.030 3.080 3.252 3.762 4.332 4.91 5.09
# 1999-03-31 2.845 2.825 2.845 3.030 3.592 4.390 4.97 5.14
# 1999-04-30 2.546 2.555 2.615 2.836 3.457 4.307 4.87 5.02
tail(xts_swap_rates, n = 4)
#                 3M     6M        1Y     2Y    5Y   10Y   20Y   30Y
# 2017-05-31 -0.3601 -0.358 -0.34700 -0.165 0.163 0.779 1.355 1.450
# 2017-06-30 -0.3586 -0.350 -0.33130 -0.130 0.267 0.895 1.444 1.531
# 2017-07-31 -0.3591 -0.351 -0.33900 -0.153 0.280 0.950 1.522 1.614
# 2017-08-31 -0.3573 -0.352 -0.34900 -0.188 0.162 0.791 1.371 1.470

# tenors of swap rates in years
tau_swap_rates <- c(1/4,1/2,1,2,5,10,20,30)
# 2D swap rates data presentation
swaprates_long <- xts_swap_rates
swaprates_long <- cbind("Date" = index(swaprates_long),as.data.frame(swaprates_long)
    )
swaprates_long <- melt(swaprates_long,"Date")

ggplot(data = swaprates_long, aes(x = Date, y = value , col = variable))+
  geom_line(size=1)+
  labs(y = "Percent (%)",x="Date" ,title="")+
  theme_bw()+
  theme_bw()+
  theme(legend.title=element_blank(),
        legend.key.width = unit(2.5,"cm"),
        axis.title = element_text(size = 32,face="plain",color="black"),
        axis.text = element_text(size = 28,face="plain",color="black"),
        legend.text = element_text(size = 29,face="plain",color="black"),
        plot.title = element_text(size = 32,face="plain",color="black"),
        legend.position = c(0.88,0.75),
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )
# Average Swap Rate Curve
```

```
swap_rates_mean <- apply(xts_swap_rates,2 ,mean )
swap_rates_mean_long <- melt(swap_rates_mean, id.vars = names(swap_rates_mean))
swap_rates_mean_long <- cbind("Tenor" = tau_swap_rates, as.data.frame(
    swap_rates_mean_long))

ggplot(data = swap_rates_mean_long, aes(x = Tenor, y = value))+
  geom_line(size=1)+
  labs(y = "Percent (%)",x="Tenor (in years)" ,title="")+
  theme_bw()+
  theme(legend.title=element_blank(),
        axis.title = element_text(size = 32,face="plain",color="black"),
        axis.text = element_text(size = 28,face="plain",color="black"),
        legend.text = element_text(size = 29,face="plain",color="black"),
        plot.title = element_text(size = 32,face="plain",color="black"),
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )

## Macroeconomic data
# macroeconomic factors stored in xts object
head(xts_macro, n = 4)
#                Inflation_Eurozone Outputgap_Eurozone EONIA_average ECB_Policy_Rate
# 1999-01-31                0.8         -0.4092192      3.137000             3.0
# 1999-02-28                0.7         -0.3684849      3.118000             3.0
# 1999-03-31                0.9         -0.3233863      2.925217             3.0
# 1999-04-30                1.1         -0.2797425      2.709545             2.5
tail(xts_macro, n = 4)
#                Inflation_Eurozone Outputgap_Eurozone EONIA_average ECB_Policy_Rate
# 2017-05-31                1.4         -0.7388219     -0.3585217            -0.4
# 2017-06-30                1.3         -0.6960000     -0.3586818            -0.4
# 2017-07-31                1.3         -0.6660192     -0.3594762            -0.4
# 2017-08-31                1.5         -0.6360384     -0.3558261            -0.4

# Graphical presentation of macroeconomic factors
data_temp <- cbind("Date" = index(xts_macro), as.data.frame(xts_macro[,c("
    Inflation_Eurozone", "Outputgap_Eurozone", "EONIA_average")]))
colnames(data_temp)[-1] <- c("Inflation","Output gap","EONIA")
data_temp <- melt(data_temp,"Date")

ggplot(data = data_temp,aes(x = Date,y = value,linetype = variable))+
  geom_line(size=1.1)+
  labs(y = "%",x="Date",title="")+
  scale_linetype_manual(values=c(1,2,4))+
  theme_bw()+
  theme(legend.title=element_blank(),
        legend.key.width = unit(1.2,"cm"),
        legend.text.align = 0,
        axis.title.x = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        axis.title.y  = element_text(size = 22,face="plain",color="black",angle = 0,
            vjust = 0.5,hjust = 0),
        legend.text = element_text(size = 20,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = c(0.8,0.85),
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )

# EONIA & ECB Policy Rate
data_temp <- cbind("Date" = index(xts_macro), as.data.frame(xts_macro[,c("
    EONIA_average","ECB_Policy_Rate")]))
colnames(data_temp)[-1] <- c("EONIA","ECB policy rate")
data_temp <- melt(data_temp,"Date")

ggplot(data = data_temp,aes(x = Date,y = value,linetype = variable))+
```

```
  geom_line(size=1.2)+
  labs(y = "",x="Date",title="")+
  scale_linetype_manual(values=c(1,2))+
  theme_bw()+
  theme(legend.title=element_blank(),
        legend.key.width = unit(1.1,"cm"),
        legend.text.align = 0,
        axis.title.x = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        axis.title.y  = element_text(size = 22,face="plain",color="black",angle = 0,
            vjust = 0.5,hjust = 0),
        legend.text = element_text(size = 20,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = c(0.8,0.9),
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )
```

A.5. Estimation methodologies

A.5.1. 2-step fix

```
##############################################
## Estimation methodologies - 2-step-fix
## set lambda values
# Nelson Siegel model
lambda0 <- 0.5978 #(maximum of curvature factor at 3Y)
# Svensson model
lambda01 <- 1.8023 #(maximum of curvature factor at 1Y)
lambda02 <- 0.5978 #(maximum of curvature factor at 3Y)

n_obs <- dim(xts_swap_rates)[1]
d_obs <- dim(xts_swap_rates)[2]
#####
# derive state variables by ordinary least squares
X <- NULL
for ( i in 1:n_obs){
  # term-structure curve at point in time i (t)
  data_temp <- as.data.frame(cbind("curve" = as.numeric(xts_swap_rates[i]),"tau" =
      tau_swap_rates))
  ## ordinary least squares - Nelson Siegel
  # temp <- lm(curve ~ slope_loading( lambda0 , tau ) + curvature_loading(lambda0,
      tau),
  #            data = data_temp)
  ## ordinary least squares - Svensson
  temp<-lm(curve~slope_loading(lambda01,tau)+curvature_loading(lambda01,tau)+
      curvature_loading(lambda02,tau),data=data_temp)
  # get estimated coefficients
  X <- rbind(X,temp$coefficients)
}
# store time series - Nelson-Siegel
# list_ts_state_variables_NS <- list()
# list_ts_state_variables_NS[["2stepfix"]] <- xts(cbind(X,rep(lambda0,times=n_obs)),
    index(xts_swap_rates))
# colnames(list_ts_state_variables_NS[["2stepfix"]]) <- c("Level","Slope","Curvature
    ", "Lambda")
# Svensson
list_ts_state_variables_Sven <- list()
list_ts_state_variables_Sven[["2stepfix"]] <- xts(cbind(X,rep(lambda01,times=n_obs),
    rep(lambda02,times=n_obs)),index(xts_swap_rates))
colnames(list_ts_state_variables_Sven[["2stepfix"]]) <- c("Level","Slope","
    Curvature1", "Curvature2","Lambda1", "Lambda2")
```

A.5.2. 2-step var

```
#################################################
## Estimation methodologies - 2-step-var using Differntial Evolution (DE)
# Parameters set by NMOF docs and own obervations
# data
data <-
 list(yM = xts_swap_rates, # swap rates as basis data
 tm = tau_swap_rates, # tenors of swap rates data
 # model = nelson_siegel_model, # or svensson_modell
 model = svensson_model,
 ww = 0.1,
 ## Restrictions, Nelson Siegel
 # restricted  based on NMOF docs # unrestricted based on NMOF docs
 min = c(0, -15,-30, 0.2561),     # min = c( 0,-15,-30,  0),
 max = c(15, 30, 30, 2.3753))    # max = c(15, 30, 30, 10))
 # Svensson
 # restr. based on 2-step est. ts          # unrestr. v1 based 2-step est. ts
 # min = c( 0,-10,-10,-10, 0.5987, 0.2561),  # min = c( 0,-10,-10,-10, 0, 0),
 # max = c(15,  5,  5,  5, 2.3753, 0.5987))   # max = c(15,  5,  5,  5,10,10))
 # unrestricted V2 based on NMOF docs
 # min = c( 0,-15,-30,-30, 0, 0),
 # max = c(15, 30, 30, 30, 10, 10))

# objective function
OF <- function(param, data) {
  y <- data$model(param, data$tm)
  residuals <- y - data$yM
  # maxdiff <- max(abs(maxdiff))
  rmse <- RMSE_fun(residuals)
  if (is.na(rmse))
    rmse <- 1e10
  rmse
}
# penalty function
penalty <- function(mP, data) { # mP matrix of Parameters
  minV <- data$min
  maxV <- data$max
  ww <- data$ww
  ## if larger than maxV, element in A is positiv
  A <- mP - as.vector(maxV)
  A <- A + abs(A)
  ## if smaller than minV, element in B is positiv
  B <- as.vector(minV) - mP
  B <- B + abs(B)
  ## beta 1 + beta2 > 0
  C <- ww*((mP[1L, ] + mP[2L, ]) - abs(mP[1L, ] + mP[2L, ]))
  A <- ww * colSums(A + B) - C
  A
}
# algorithmus
algo <-
  list(nP = 200,   ## number of populations
  nG = 1000L,   ## number of generations
  F = 0.5,   ## step size
  CR = 0.99,   ## prob. of crossover
  ## Restrictions - Nelson-Siegel
  # restricted  based on NMOF docs   # unrestricted based on NMOF docs
  min = c(0, -15,-30, 0.2561),        # min = c( 0,-15,-30,  0),
  max = c(15, 30, 30, 2.3753),        # max = c(15, 30, 30, 10),
  # Svensson
  # restr. based on 2-step est. ts         # unrestr. v1 based 2-step est. ts
  # min = c( 0,-10,-10,-10, 0.5987, 0.2561), # min = c( 0,-10,-10,-10, 0, 0),
```

```
  # max = c(15,  5,  5,  5, 2.3753, 0.5987),  # max = c(15,  5,  5,  5,10,10),
  # unrestricted V2 based on NMOF docs
  # min = c( 0,-15,-30,-30, 0, 0),
  # max = c(15, 30, 30, 30, 10, 10),
  pen = penalty,
  repair = NULL,
  loopOF = TRUE,   ## loop over popuation - yes
  loopPen = FALSE,  ## loop over penalty -  no
  loopRepair = TRUE,   ## loop over popuation -  yes
  printBar = FALSE)

X <- NULL
for ( i in as.character(index(xts_swap_rates))){
    data$yM <- xts_swap_rates[i]
    sol <- DEopt(OF = OF, algo = algo, data = data)
    X <- rbind(X,sol$xbest)
}
# store time series
list_ts_state_variables_NS[["2stepvar"]] <- xts(X,index(xts_swap_rates))
colnames(list_ts_state_variables_NS[["2stepvar"]]) <- c("Level","Slope","Curvature",
     "Lambda")
# list_ts_state_variables_Sven[["2stepvar_unr"]] <- xts(X,index(xts_swap_rates))
# colnames(list_ts_state_variables_Sven[["2stepvar_unr"]]) <- c("Level","Slope","
    Curvature1", "Curvature2","Lambda1", "Lambda2")
```

A.5.3. 1-step fix

```
###############################################
## Estimation methodologies - 1-step
## preparation for one-step estimation method defined as function
## Nelson-Siegel
# Initial values based on two-step estimation approach
prep_1step_NS <- function(lambda0, xts_swap_rates_temp){
 ## Two step estimation method
 X <- NULL
 for ( i in 1:n_obs){
  data_temp <- as.data.frame(cbind("curve" = as.numeric(xts_swap_rates_temp[i]),
                                   "tau" = tau_swap_rates))
  temp <- lm(curve ~ slope_loading(lambda0,tau )+curvature_loading(lambda0,tau),
             data = data_temp)
  X <- rbind(X,
             temp$coefficients)
  }
  X <- cbind(X,rep(lambda0,times=n_obs))
  X <- xts(X,index(xts_swap_rates_temp))
  colnames(X) <- c("Level","Slope","Curvature", "Lambda")

  ## 1. In order to identify plausible starting values of the parameters to be
     estimated the information of the two-step estimation approach is used.
  # VAR
  VAR2step <- VAR(X[,c("Level","Slope","Curvature")], 1, "const") #  Nelson-Siegel
 # Residuals of estimated yields 2-step
  C <- matrix(c(rep(1, d_obs),
                slope_loading(lambda0,tau_swap_rates),
                curvature_loading(lambda0, tau_swap_rates)), ncol = 3, nrow = d_obs)
  est_yields <- t(C %*% t(X[,c("Level","Slope","Curvature")]))
  residuals <- est_yields - as.data.frame(xts_swap_rates_temp)

  ## 2. General initial values preparation - initialize the starting values and mean
      adjusted yields and factors (X_t)
  mu0 <- apply(X[,c("Level","Slope","Curvature")],2,function(x) mean(x))
  A0 <- Bcoef(VAR2step)[,-4]
```

```
  Q0 <- summary(VAR2step)
  Q0 <- Q0$covres
  B0 <- matrix(0, ncol = 3, nrow = 3)
  diag(B0) <- sqrt(diag(Q0))
  H0 <- cov(residuals)
  D0 <- matrix(0, ncol  = d_obs, nrow = d_obs)
  diag(D0) <- sqrt(diag(H0))
  x0 <- as.numeric(X[1,c("Level","Slope","Curvature")])
  z0 <- x0 - mu0
  P0 <- cov(X[,c("Level","Slope","Curvature")])

  ## Vector parametrization of initial values
  A0_v <- as.vector(A0)
  names(A0_v) <- paste0("a", 1:9)
  B0_v <- as.vector(B0)
  # for B0 only lower triangle Matrix
  B0_v <- B0_v[c(1:3, 5:6, 9)]
  names(B0_v) <- paste0("b", 1:6)
  D0_v <- diag(D0)
  names(D0_v) <- paste0("d",1:8)
  names(mu0) <- paste0("m", 1:3)
  names(lambda0) <- "lambda"
  ## Initial values in list
  list_initial_values <- c(as.list(A0_v),as.list(B0_v),as.list(D0_v),as.list(mu0),
      as.list(lambda0))
  list("P0" = P0, "x0" = x0, "initial_values" = list_initial_values)
}

## 3. Kalman Filter and MLE
## FKF Package
dt <- matrix(0,3,1)
ct = matrix(0,d_obs,1)
y_t <- t(xts_swap_rates_temp)
## State space model FKF
SS_FKF_LL_1st_NS <- function(a1,a2,a3,a4,a5,a6,a7,a8,a9,
                             b1,b2,b3,b4,b5,b6,
                             d1, d2, d3, d4, d5, d6, d7, d8,
                             m1, m2, m3,
                             lambda){
  Tt <- matrix(c(a1,a2,a3,a4,a5,a6,a7,a8,a9), ncol = 3, nrow = 3)
  Ht <- matrix(c(b1,b2,b3,0,b4,b5,0,0,b6), ncol = 3, nrow = 3)
  HHt <- Ht %*% t(Ht)
  Zt <- matrix(c(rep(1, d_obs),
                 slope_loading(lambda,tau_swap_rates),
                 curvature_loading(lambda, tau_swap_rates)), ncol = 3, nrow = d_obs)
  a0 <- x0 - c(m1, m2, m3)
  mu0_yields <- matrix(Zt %*% c(m1, m2, m3), nrow = d_obs, ncol = n_obs)
  y_t_deflated = y_t - mu0_yields
  Gt <- diag(c(d1, d2, d3, d4, d5, d6, d7, d8))
  GGt <- Gt %*% t(Gt)
  #print(lambda)
    SS_FKF <- fkf(a0 = a0, P0 = P0, dt = dt, ct = ct, Tt = Tt, Zt = Zt, HHt = HHt,
        GGt = GGt, yt = y_t_deflated, check.input = T)
  - SS_FKF$logLik
}
## MLE - FKF
lambda0 <- as.numeric(list_ts_state_variables_NS$‘2stepfix‘[1,c("Lambda")])
xts_swap_rates_temp <- xts_swap_rates
n_obs <- dim(xts_swap_rates_temp)[1]
d_obs <- dim(xts_swap_rates_temp)[2]
initial_values <- prep_1step_NS(lambda0, xts_swap_rates_temp)
list_initial_values <- initial_values$initial_values
P0 <- initial_values$P0
```

```
x0 <- initial_values$x0
library(stats4)
## Maximum Likelihood estimation using algorithm BFGS
NS_MLE_BFGS_fkf<-mle(SS_FKF_LL_lst_NS,start=list_initial_values,method="BFGS")

## LU version of BFGS
## fkf with lower and upper values
lower_NS <- c(-Inf, -Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf, -Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf, -Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf, -Inf,-Inf,
           0.01)
upper_NS <- c(Inf, Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,
           Inf, Inf,Inf,Inf,Inf,Inf,
           Inf, Inf,Inf,Inf,Inf,Inf,Inf,Inf,
           Inf, Inf,Inf,
           3)
names(lower_NS) <- names(upper_NS) <- names(list_initial_values)
## Maximum Likelihood estimation using algorithm Limited-memory-BFGS
NS_MLE_LU_BFGS_fkf<-mle(SS_FKF_LL_lst_NS,start=list_initial_values,method="L-BFGS-B"
    , lower = lower_NS, upper = upper_NS)

## dlm package
#####
library(dlm)
## State space model dlm
SS_DLM_LL_lst_NS <- function(a1,a2,a3,a4,a5,a6,a7,a8,a9,
                             b1,b2,b3,b4,b5,b6,
                             d1, d2, d3, d4, d5, d6, d7, d8,
                             m1, m2, m3,
                             lambda){
  GG <- matrix(c(a1,a2,a3,a4,a5,a6,a7,a8,a9), ncol = 3, nrow = 3)
  W <- matrix(c(b1,b2,b3,0,b4,b5,0,0,b6), ncol = 3, nrow = 3)
  W <- W %*% t(W)
  FF <- matrix(c(rep(1, d_obs),
                 slope_loading(lambda,tau_swap_rates),
                 curvature_loading(lambda, tau_swap_rates)), ncol = 3, nrow = d_obs)
  m0 <- x0 - c(m1, m2, m3)
  mu0_yields <- matrix(t(FF %*% c(m1, m2, m3)),ncol=d_obs,nrow=n_obs,byrow=T)
  y_t_deflated = xts_swap_rates_temp - mu0_yields
  V <- diag(c(d1, d2, d3, d4, d5, d6, d7, d8))
  V <- V %*% t(V)
  #print(lambda)
  SS_dlm <- dlm(FF = FF, GG = GG, V = V, W = W, m0 = m0, C0 = P0)
  dlmLL(y_t_deflated, SS_dlm)
}
## not working due to negative/extreme lambda value as starting values in one
    iteration
# NS_MLE_BFGS_dlm<-mle(SS_DLM_LL_lst_NS,start=list_initial_values,method="BFGS")
## LU version of BFGS
NS_MLE_LU_BFGS_dlm<-mle(SS_DLM_LL_lst_NS,start=list_initial_values,method="L-BFGS-B"
    ,lower = lower_NS,upper = upper_NS)
#####
list_KF_coef_NS <- list("FKF" = coef(NS_MLE_BFGS_fkf),
                        "FKF_LU" = coef(NS_MLE_LU_BFGS_fkf),
                        "dlm_LU" = coef(NS_MLE_LU_BFGS_dlm))

## Derive smoothed state variables
#####
# Estimated coefficients, smoothing by dlm package
# state space set-up
SS_DLM_NS <- function(params){
  GG <- matrix(c(params[1:9]), ncol = 3, nrow = 3)
```

```
  W <- matrix(c(params[10:12],0,params[13:14],0,0,params[15]), ncol = 3, nrow = 3)
  W <- W %*% t(W)
  FF <- matrix(c(rep(1, d_obs),
                 slope_loading(params[27],tau_swap_rates),
                 curvature_loading(params[27], tau_swap_rates)),ncol=3,nrow=d_obs)
  m0 <- x0 - params[24:26]
  V <- diag(c(params[16:23]))
  V <- V %*% t(V)
  SS_dlm <- dlm(FF = FF, GG = GG, V = V, W = W, m0 = m0, C0 = P0)
  SS_dlm
}
# define state space model by estimated parameters
SS_presentations_dlm <- list()
for ( i in names(list_KF_coef_NS)){
  SS_presentations_dlm[[i]] <- SS_DLM_NS(list_KF_coef_NS[[i]])
}

# smooth dynamics of state variables
list_est_smoothed_states_NS <- list()
for ( i in names(list_KF_coef_NS)){
  mu0_yields <- matrix(t(FF(SS_presentations_dlm[[i]]) %*% list_KF_coef_NS[[i]][c("m1","m2","m3")]), ncol = d_obs, nrow = n_obs, byrow = T)
  y_t_deflated = xts_swap_rates - mu0_yields
  list_est_smoothed_states_NS[[i]] <- dlmSmooth(y_t_deflated,SS_presentations_dlm[[i]])$s[-1,] + matrix(list_KF_coef_NS[[i]][c("m1","m2","m3")],ncol = 3, nrow = n_obs, byrow = T)
  list_est_smoothed_states_NS[[i]] <- cbind(list_est_smoothed_states_NS[[i]],
                                     rep(list_KF_coef_NS[[i]][c("lambda")],224))
 colnames(list_est_smoothed_states_NS[[i]])<-c("Level","Slope","Curvature","Lambda")
 list_est_smoothed_states_NS[[i]]<-xts(list_est_smoothed_states_NS[[i]],index(xts_swap_rates))
}
## used for DA
list_ts_state_variables_NS[["1step"]] <- list_est_smoothed_states_NS$dlm_LU

#####
## Svensson
#####
prep_1step_Sven <- function(lambda01,lambda02, xts_swap_rates_temp){
  ## Two step estimation method
  X <- NULL
  for ( i in 1:n_obs){
    data_temp <- as.data.frame(cbind("curve" = as.numeric(xts_swap_rates_temp[i]),
                                     "tau" = tau_swap_rates))
    temp <- lm(curve ~ slope_loading( lambda01 , tau ) + curvature_loading(lambda01, tau) + curvature_loading(lambda02, tau),
               data = data_temp)
    X <- rbind(X,
               temp$coefficients)
  }
  X <- cbind(X,rep(lambda01,times=n_obs),rep(lambda01,times=n_obs))
  X <- xts(X,index(xts_swap_rates_temp))
  colnames(X) <- c("Level","Slope","Curvature1", "Curvature2","Lambda1","Lambda2")

  ## 1. In order to identify plausible starting values of the parameters to be estimated the information of the two-step estimation approach is used.
  # VAR
  VAR2step <- VAR(X[,c("Level","Slope","Curvature1","Curvature2")], 1, "const")
  # residuals
  C <- matrix(c(rep(1, d_obs),
                slope_loading(lambda01,tau_swap_rates),
                curvature_loading(lambda01, tau_swap_rates),
                curvature_loading(lambda02, tau_swap_rates)),ncol=4,nrow=d_obs)
```

```
  est_yields <- t(C %*% t(X[,c("Level","Slope","Curvature1","Curvature2")]))
  residuals <- est_yields - as.data.frame(xts_swap_rates_temp)

 ## 2. General initial values preparation - initialize the starting values and mean adjusted yields and factors (X_t) according to Diebold & Li
  mu0<-apply(X[,c("Level","Slope","Curvature1","Curvature2")],2,function(x) mean(x))
  A0 <- Bcoef(VAR2step)[,-5]
  Q0 <- summary(VAR2step)
  Q0 <- Q0$covres
  B0 <- matrix(0, ncol = 4, nrow = 4)
  diag(B0) <- sqrt(diag(Q0))
  H0 <- cov(residuals)
  D0 <- matrix(0, ncol  = d_obs, nrow = d_obs)
  diag(D0) <- sqrt(diag(H0))
  x0 <- as.numeric(X[1,c("Level","Slope","Curvature1","Curvature2")])
  z0 <- x0 - mu0
  P0 <- cov(X[,c("Level","Slope","Curvature1","Curvature2")])

  ## Vector parametrization of initial values
  A0_v <- as.vector(A0)
  names(A0_v) <- paste0("a", 1:16)
  B0_v <- as.vector(B0)
  # for B0 only lower triangle Matrix
  B0_v <- B0_v[c(1:4, 6:8, 11:12, 16)]
  names(B0_v) <- paste0("b", 1:10)
  D0_v <- diag(D0)
  names(D0_v) <- paste0("d",1:8)
  names(mu0) <- paste0("m", 1:4)
  #lambda0 <- log(lambda0)
  names(lambda01) <- "lambda1"
  names(lambda02) <- "lambda2"

  list_initial_values <- c(as.list(A0_v),as.list(B0_v),as.list(D0_v),as.list(mu0),as.list(lambda01),as.list(lambda02))
  list("P0" = P0, "x0" = x0, "initial_values" = c(as.list(A0_v),as.list(B0_v),as.list(D0_v),as.list(mu0),as.list(lambda01),as.list(lambda02)))
}

## 3. Kalman Filter and MLE
## FKF Package
dt <- matrix(0,4,1)
ct = matrix(0,d_obs,1)
y_t <- t(xts_swap_rates)

SS_FKF_LL_1st_Sven<-function(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,
                                b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,
                                d1, d2, d3, d4, d5, d6, d7, d8,
                                m1, m2, m3,m4,
                                lambda1,lambda2){
  Tt<-matrix(c(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16),ncol=4,nrow=4)
  Ht <- matrix(c(b1,b2,b3,b4,0,b5,b6,b7,0,0,b8,b9,0,0,0,b10), ncol = 4, nrow = 4)
  HHt <- Ht %*% t(Ht)
  Zt <- matrix(c(rep(1, d_obs),
                 slope_loading(lambda1,tau_swap_rates),
                 curvature_loading(lambda1, tau_swap_rates),
                 curvature_loading(lambda2, tau_swap_rates)),ncol=4, nrow=d_obs)
  a0 <- x0 - c(m1, m2, m3, m4)
  mu0_yields <- matrix(Zt %*% c(m1, m2, m3, m4), nrow = d_obs, ncol = n_obs)
  y_t_deflated = y_t - mu0_yields
  Gt <- diag(c(d1, d2, d3, d4, d5, d6, d7, d8))
  GGt <- Gt %*% t(Gt)
  # print(lambda1)
```

```
  # print(lambda2)

  SS_FKF <- fkf(a0 = a0, P0 = P0, dt = dt, ct = ct, Tt = Tt, Zt = Zt, HHt = HHt, GGt = GGt, yt = y_t_deflated, check.input = T)

  - SS_FKF$logLik
}

lambda01<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("Lambda1")]) #(1Y)
lambda02<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("Lambda2")]) #(3Y)
xts_swap_rates_temp <- xts_swap_rates
n_obs <- dim(xts_swap_rates_temp)[1]
d_obs <- dim(xts_swap_rates_temp)[2]
initial_values <- prep_1step_Sven(lambda01,lambda02, xts_swap_rates_temp)
list_initial_values <- initial_values$initial_values
P0 <- initial_values$P0
x0 <- initial_values$x0

MLE_BFGS_fkf_Sven<-mle(SS_FKF_LL_1st_Sven,start=list_initial_values,method="BFGS")

## LU version of BFGS
lower_Sven<-c(-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,-Inf,
           -Inf,-Inf,-Inf,-Inf,
           0.01,0.01)
upper_Sven<-c(Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,
           Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,
           Inf,Inf,Inf,Inf,Inf,Inf,Inf,Inf,
           Inf,Inf,Inf,Inf,
           3,3)
names(lower_Sven) <- names(upper_Sven) <- names(list_initial_values)

MLE_LU_BFGS_fkf_Sven<-mle(SS_FKF_LL_1st_Sven,start=list_initial_values,method="L-BFGS-B",lower = lower_Sven,upper = upper_Sven)

## dlm package
SS_DLM_LL_1st_Sven<-function(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,
                                b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,
                                d1, d2, d3, d4, d5, d6, d7, d8,
                                m1, m2, m3,m4,
                                lambda1,lambda2){
  GG<-matrix(c(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16),ncol=4,nrow=4)
  W <- matrix(c(b1,b2,b3,b4,0,b5,b6,b7,0,0,b8,b9,0,0,0,b10), ncol = 4, nrow = 4)
  W <- W %*% t(W)
  FF <- matrix(c(rep(1, d_obs),
                 slope_loading(lambda1,tau_swap_rates),
                 curvature_loading(lambda1, tau_swap_rates),
                 curvature_loading(lambda2, tau_swap_rates)),ncol=4,nrow=d_obs)
  m0 <- x0 - c(m1, m2, m3, m4)
  mu0_yields<-matrix(t(FF %*% c(m1, m2, m3, m4)),ncol=d_obs,nrow=n_obs,byrow=T)
  y_t_deflated = xts_swap_rates_temp - mu0_yields
  V <- diag(c(d1, d2, d3, d4, d5, d6, d7, d8))
  V <- V %*% t(V)
  #print(c(lambda1,lambda2))
  SS_dlm <- dlm(FF = FF, GG = GG, V = V, W = W, m0 = m0, C0 = P0)
  dlmLL(y_t_deflated, SS_dlm)
}
## BFGS
## not working due to negative/extreme lambda value as starting values in one iteration
```

```
# MLE_BFGS_dlm_Sven<-mle(SS_DLM_LL_lst_Sven,start=list_initial_values,method="BFGS")
## L-BFGS-B
MLE_LU_BFGS_dlm_Sven<-mle(SS_DLM_LL_lst_Sven,start=list_initial_values,method="
    L-BFGS-B",lower = lower_Sven,upper = upper_Sven)

list_KF_coef_Sven <- list("FKF" = coef(MLE_BFGS_fkf_Sven),
                          "FKF_LU" = coef(MLE_LU_BFGS_fkf_Sven),
                          "dlm_LU" = coef(MLE_LU_BFGS_dlm_Sven))

#####
## Derive smoothed state variables
# Estimated coefficients, smoothing by dlm package
# state space set-up
SS_DLM_Sven <- function(params){
  GG <- matrix(c(params[1:16]), ncol = 4, nrow = 4)
  W<-matrix(c(params[17:20],0,params[21:23],0,0,params[24:25],0,0,0,params[26]),ncol
      =4,nrow=4)
  W <- W %*% t(W)
  FF <- matrix(c(rep(1, d_obs),
                 slope_loading(params[39],tau_swap_rates),
                 curvature_loading(params[39], tau_swap_rates),
                 curvature_loading(params[40], tau_swap_rates)),ncol=4,nrow=d_obs)
  m0 <- x0 - params[35:38]
  V <- diag(c(params[27:34]))
  V <- V %*% t(V)
  SS_dlm <- dlm(FF = FF, GG = GG, V = V, W = W, m0 = m0, C0 = P0)
  SS_dlm
}
## state space model by estimated parameters
SS_presentations_dlm_Sven <- list()
for ( i in names(list_KF_coef_Sven)){
  SS_presentations_dlm_Sven[[i]] <- SS_DLM_Sven(list_KF_coef_Sven[[i]])
}
## derive smoothed state variables
list_est_smoothed_states_Sven <- list()
for ( i in names(list_KF_coef_Sven)){
  mu0_yields <- matrix(t(FF(SS_presentations_dlm_Sven[[i]]) %*% list_KF_coef_Sven[[i
      ]][c("m1","m2","m3","m4")]), ncol = d_obs, nrow = n_obs, byrow = T)
  y_t_deflated = xts_swap_rates - mu0_yields

  list_est_smoothed_states_Sven[[i]] <- dlmSmooth(y_t_deflated,
      SS_presentations_dlm_Sven[[i]])$s[-1,] + matrix(list_KF_coef_Sven[[i]][c("m1",
      "m2","m3","m4")],ncol = 4, nrow = n_obs, byrow = T)
  list_est_smoothed_states_Sven[[i]] <- cbind(list_est_smoothed_states_Sven[[i]],
                                    rep(list_KF_coef_Sven[[i]][c("lambda1")],224),
                                    rep(list_KF_coef_Sven[[i]][c("lambda2")],224))
  colnames(list_est_smoothed_states_Sven[[i]]) <- c("Level","Slope","Curvature1","
      Curvature2","Lambda1","Lambda2")
  list_est_smoothed_states_Sven[[i]] <- xts(list_est_smoothed_states_Sven[[i]],
      index(xts_swap_rates))
}
## used for DA
list_ts_state_variables_Sven[["1step"]] <- list_est_smoothed_states_Sven[["dlm_LU"]]
```

A.5.4. 1-2-step fix

```
###############################################
## Estimation methodologies - 1-2-step-fix
## set lambda values as estimated by one-step estimation approach
# Nelson Siegel model
lambda0 <- as.numeric(list_ts_state_variables_NS[["1step"]][1,"Lambda"])
# Svensson model
```

```
lambda01 <- as.numeric(list_ts_state_variables_Sven[["1step"]][1,"Lambda1"])
lambda02 <- as.numeric(list_ts_state_variables_Sven[["1step"]][1,"Lambda2"])
n_obs <- dim(xts_swap_rates)[1]
d_obs <- dim(xts_swap_rates)[2]
#####
# derive state variables by ordinary least squares
X <- NULL
for ( i in 1:n_obs){
  # term-structure curve at point in time i (t)
  data_temp <- as.data.frame(cbind("curve" = as.numeric(xts_swap_rates[i]),
                                    "tau" = tau_swap_rates))
  ## ordinary least squares - Nelson Siegel
  temp<-lm(curve~slope_loading(lambda0,tau)+curvature_loading(lambda0,tau),data =
      data_temp)
  ## ordinary least squares - Svensson
  # temp<-lm(curve~slope_loading(lambda01,tau)+curvature_loading(lambda01,tau)+
      curvature_loading(lambda02,tau),
  #            data = data_temp)
  # get estimated coefficients
  X <- rbind(X,temp$coefficients)
}
# store time series - Nelson-Siegel
list_ts_state_variables_NS[["12step"]] <- xts(cbind(X,rep(lambda0,times=n_obs)),
    index(xts_swap_rates))
colnames(list_ts_state_variables_NS[["12step"]]) <- c("Level","Slope","Curvature", "
    Lambda")
# Svensson
# list_ts_state_variables_Sven[["12step"]] <- xts(cbind(X,rep(lambda01,times=n_obs),
    rep(lambda02,times=n_obs)),index(xts_swap_rates))
# colnames(list_ts_state_variables_Sven[["12step"]]) <- c("Level","Slope","
    Curvature1", "Curvature2","Lambda1", "Lambda2")
```

A.6. In-sample fit

A.6.1. In-sample fit results

The following code includes: Table 5.1, Table 5.2, Figure 5.1, Figure 5.2, Figure 5.3

```
#####
## Overall in-sample fit and dynamics in the term-structure
# calculation of estimated yields by various model specifications
estimated_yields <- list()
# Nelson-Siegel/Svensson - fix decay parameters
str_temp <- c("2stepfix", "12step", "1step")
for ( i in str_temp){
  # Nelson-Siegel
  estimated_yields[[paste0("NS_",i)]]< list_ts_state_variables_NS[[i]][,c("Level","
      Slope","Curvature")] %*% rbind(rep(1,d_obs),slope_loading(as.numeric(
      list_ts_state_variables_NS[[i]][1,"Lambda"]),tau_swap_rates),
      curvature_loading(as.numeric(list_ts_state_variables_NS[[i]][1,"Lambda"]),
      tau_swap_rates))
  estimated_yields[[paste0("NS_",i)]]<-xts(estimated_yields[[paste0("NS_",i)]],index
      (xts_swap_rates))
  # Svensson
  estimated_yields[[paste0("Sven_",i)]] <- list_ts_state_variables_Sven[[i]][,c("
      Level","Slope","Curvature1", "Curvature2")] %*% rbind(rep(1, d_obs),
      slope_loading(as.numeric(list_ts_state_variables_Sven[[i]][1,"Lambda1"]),
      tau_swap_rates),curvature_loading(as.numeric(list_ts_state_variables_Sven[[i
      ]][1,"Lambda1"]),tau_swap_rates),curvature_loading(as.numeric(
      list_ts_state_variables_Sven[[i]][1,"Lambda2"]),tau_swap_rates))
```

```
  estimated_yields[[paste0("Sven_",i)]] <- xts(estimated_yields[[paste0("Sven_",i)]], index(xts_swap_rates))
}
# Nelson-Siegel/svensson - var decay parameters
str_temp <- "2stepvar"
estimated_yields[[paste0("NS_",str_temp)]] <- NULL
estimated_yields[[paste0("Sven_",str_temp)]] <- NULL
for ( i in 1:dim(xts_swap_rates)[1]){
  # Nelson-Siegel
  estimated_yields[[paste0("NS_",str_temp)]] <- rbind(estimated_yields[[paste0("NS_",str_temp)]],nelson_siegel_model(as.numeric(list_ts_state_variables_NS[[str_temp]][i,]),tau_swap_rates))
  # Svensson
  estimated_yields[[paste0("Sven_",str_temp)]] <- rbind(estimated_yields[[paste0("Sven_",str_temp)]],svensson_model(as.numeric(list_ts_state_variables_Sven[[str_temp]][i,]),tau_swap_rates))
}
estimated_yields[[paste0("NS_",str_temp)]] <-  xts(estimated_yields[[paste0("NS_",str_temp)]], index(xts_swap_rates))
estimated_yields[[paste0("Sven_",str_temp)]] <- xts(estimated_yields[[paste0("Sven_",str_temp)]], index(xts_swap_rates))
## calculation of residuals
str_temp <- c("NS_2stepfix", "NS_2stepvar", "NS_1step", "NS_12step", "Sven_2stepfix", "Sven_2stepvar", "Sven_1step", "Sven_12step")
residuals <- list()
for ( i in str_temp){
  residuals[[i]] <- estimated_yields[[i]] - xts_swap_rates
}

## calculation of measure to examine the in-sample fit of the models
# table - RMSE over term-structure curve - change 2-step var in DA
table_termstructure(residuals, names(residuals))
# [1] "1999-01-31"
# [1] "2017-08-31"
#                  Mean Median Std. Dev.   Min.   Max.
# NS_2stepfix   0.0752 0.0682     0.0386 0.0274 0.2730
# NS_2stepvar   0.0640 0.0568     0.0324 0.0194 0.2055
# ....

# table - RMSE per individual tenors
table_tenor(residuals, names(residuals))
# $RMSE
#  NS_2stepfix NS_2stepvar NS_1step NS_12step Sven_2stepfix Sven_2stepvar Sven_1step Sven_12step
# 3M    0.0722       0.0420   0.0949    0.0828         0.0165         0.0226     0.0031     0.0366
# 6M    0.0204       0.0292   0.0000    0.0194         0.0247         0.0391     0.0140     0.0175
# ....

# figure - mean per individual tenors
str_labels_plot <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
colours <- blue2yellow(n = 5)[c(1,2,3,4)]
names(colours) <- c("X2.step.fix","X2.step.var","X1.step.2.step","X1.step")
str_labels <- c("2-step fix", "2-step var","1-step","1-2-step")
names(str_labels) <- str_labels_plot
# data
data_temp <- table_tenor(residuals, names(residuals))$Mean
# Nelson-Siegel
data_temp_fit_ns <- data_temp[,1:4]
colnames(data_temp_fit_ns) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_fit_ns <- as.data.frame(cbind("Tenor" = tau_swap_rates,data_temp_fit_ns))
```

```
data_temp_fit_ns <- melt(data_temp_fit_ns , "Tenor")
data_temp_fit_ns <- cbind(data_temp_fit_ns,"Model" = rep("Nelson-Siegel",dim(data_temp_fit_ns)[1]))
colnames(data_temp_fit_ns)[2] <- "Method"
# Svensson
data_temp_fit_sven <- data_temp[,5:8]
colnames(data_temp_fit_sven) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_fit_sven <- as.data.frame(cbind("Tenor" = tau_swap_rates,data_temp_fit_sven))
data_temp_fit_sven <- melt(data_temp_fit_sven , "Tenor")
data_temp_fit_sven <- cbind(data_temp_fit_sven,"Model" = rep("Svensson",dim(data_temp_fit_sven)[1]))
colnames(data_temp_fit_sven)[2] <- "Method"
# Combine
data_temp <- rbind(data_temp_fit_ns,data_temp_fit_sven)
# Plot
ggplot(data = data_temp,aes(x = Tenor,y = value,linetype = Model, col = Method))+
  geom_line(size=1.2)+
  labs(y = "Mean",x="Tenor (in years)",title="")+
  #  scale_linetype_manual(labels = str_models_temp,values=c(1,2,4,5,6,7))+
  scale_color_manual(labels = str_labels, values = colours)+
  guides(color = guide_legend(order = 2), linetype = guide_legend(order = 1)) +
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.spacing.x = unit(0.1,"cm"),
        legend.margin = margin(0, 0, 0, 0, "cm"),
        legend.key.width = unit(1,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 21,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 20,face="plain",color="black"),
        legend.position = "right",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )

# figure - RSME over term-structure evolved over time
rmse_swapratecurve <- NULL
for ( j in 1:length(residuals)){
  rmse_swapratecurve <- cbind(rmse_swapratecurve,apply(residuals[[j]],1,RMSE_fun))
}
data_temp <- cbind("Date" = index(xts_swap_rates),as.data.frame(rmse_swapratecurve))
colnames(data_temp)[-1] <- names(residuals)
# Nelson-Siegel
data_temp_fit_ns <- data_temp[,1:5]
colnames(data_temp_fit_ns)[-1] <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_fit_ns <- melt(data_temp_fit_ns , "Date")
data_temp_fit_ns <- cbind(data_temp_fit_ns,"Model" = rep("Nelson-Siegel",dim(data_temp_fit_ns)[1]))
colnames(data_temp_fit_ns)[2] <- "Method"
# Svensson
data_temp_fit_sven <- data_temp[,c(1,6:9)]
colnames(data_temp_fit_sven)[-1] <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_fit_sven <- melt(data_temp_fit_sven , "Date")
data_temp_fit_sven <- cbind(data_temp_fit_sven,"Model" = rep("Svensson",dim(data_temp_fit_sven)[1]))
colnames(data_temp_fit_sven)[2] <- "Method"
data_temp <- rbind(data_temp_fit_ns,data_temp_fit_sven)

ggplot(data = data_temp,aes(x = Date,y = value,linetype = Model, col = Method))+
```

```
  geom_line(size=1.2)+
  labs(y = "RMSE",x="Date",title="")+
  #  scale_linetype_manual(labels = str_models_temp,values=c(1,2,4,5,6,7))+
  scale_color_manual(labels = str_labels, values = colours)+
  guides(color = guide_legend(order = 2), linetype = guide_legend(order = 1)) +
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.spacing.x = unit(0.1,"cm"),
        legend.margin = margin(0, 0, 0, 0, "cm"),
        legend.key.width = unit(1,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 21,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 20,face="plain",color="black"),
        legend.position = c(0.12,0.72),
        plot.margin = unit(c(0,0.1,0,0),"cm")

# figure - shapes of the term-structure and average shape of the term-structure
# time points
#  - Average swap rate curve
#  - 01/2002 & 06/2005 & 08/2008 & 10/2011 & 01/2017
time_stamps <- c("2002-01","2005-06","2008-08","2011-10","2017-01")
Sys.setlocale("LC_TIME","English")
time_string <- format(as.Date(paste(time_stamps,"01",sep= "-"),"%Y-%m-%d"),"%b, %Y")
k <- 1
list_plots <- NULL
str_temp <- c("NS_2stepfix", "NS_2stepvar", "NS_1step", "NS_12step", "Sven_2stepfix", "Sven_2stepvar", "Sven_1step", "Sven_12step")
for ( i in time_stamps){
  data_temp_act <- data.frame("Tenor" = tau_swap_rates,"Actual" = as.numeric(xts_swap_rates[i,]))
  data_temp_act <- melt(data_temp_act, "Tenor")
  data_temp_ns <- NULL
  # Nelson-Siegel
  for ( j in str_temp[1:4]){
    data_temp_ns <- cbind(data_temp_ns,as.numeric(estimated_yields[[j]][i,]))
  }
  colnames(data_temp_ns) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
  data_temp_ns <- data.frame("Tenor" = tau_swap_rates,data_temp_ns)
  data_temp_ns <- melt(data_temp_ns, "Tenor")
  data_temp_ns <- cbind(data_temp_ns,"Model" = rep("Nelson-Siegel",dim(data_temp_ns)[1]))
  colnames(data_temp_ns)[2] <- "Method"
  data_temp_sven <- NULL
  # Svensson
  for ( j in str_temp[5:8]){
    data_temp_sven <- cbind(data_temp_sven,as.numeric(estimated_yields[[j]][i,]))
  }
  colnames(data_temp_sven) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
  data_temp_sven <- data.frame("Tenor" = tau_swap_rates,data_temp_sven)
  data_temp_sven <- melt(data_temp_sven, "Tenor")
  data_temp_sven <- cbind(data_temp_sven,"Model" = rep("Svensson",dim(data_temp_sven)[1]))
  colnames(data_temp_sven)[2] <- "Method"
  data_temp <- rbind(data_temp_ns,data_temp_sven)

  list_plots[[i]] <- ggplot() +
    geom_line(data = data_temp, mapping = aes(x = Tenor,y = value, linetype = Model, color = Method), size = 1.2) +
```

```
    geom_point(data = data_temp_act, mapping = aes(x = Tenor,y = value, shape = variable), size = 2) +
    scale_color_manual(labels = str_labels, values = colours)+
    labs(y = "",x="Date",title=time_string[k], shape = "") +
    guides(shape = guide_legend(order = 1),color = guide_legend(order = 3, ncol = 2), linetype = guide_legend(order = 2)) +
    theme_bw()+
    theme(legend.title=element_text(size = 17,face="plain",color="black"),
          legend.key.width = unit(1.2,"cm"),
          legend.text.align = 0,
          axis.title = element_text(size = 18,face="plain",color="black"),
          axis.text = element_text(size = 18,face="plain",color="black"),
          legend.text = element_text(size = 15,face="plain",color="black"),
          plot.title = element_text(size = 19,face="plain",color="black"),
          #legend.position = c(0.68,0.43),
          legend.position = "none",
          plot.margin = unit(c(0,0.1,0,0),"cm")
    )
  k <- k + 1
}
## average
data_temp_act <- data.frame("Tenor" = tau_swap_rates,"Actual" = apply(xts_swap_rates,2,mean))
data_temp_act <- melt(data_temp_act, "Tenor")
data_temp_ns <- NULL
# Nelson-Siegel
for ( j in str_temp[1:4]){
  data_temp_ns <- cbind(data_temp_ns,apply(estimated_yields[[j]],2,mean))
}
colnames(data_temp_ns) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_ns <- data.frame("Tenor" = tau_swap_rates,data_temp_ns)
data_temp_ns <- melt(data_temp_ns, "Tenor")
data_temp_ns <- cbind(data_temp_ns,"Model" = rep("Nelson-Siegel",dim(data_temp_ns)[1]))
colnames(data_temp_ns)[2] <- "Method"
data_temp_sven <- NULL
# Svensson
for ( j in str_temp[5:8]){
  data_temp_sven <- cbind(data_temp_sven,apply(estimated_yields[[j]],2,mean))
}
colnames(data_temp_sven) <- c("X2.step.fix","X2.step.var","X1.step","X1.step.2.step")
data_temp_sven <- data.frame("Tenor" = tau_swap_rates,data_temp_sven)
data_temp_sven <- melt(data_temp_sven, "Tenor")
data_temp_sven <- cbind(data_temp_sven,"Model" = rep("Svensson",dim(data_temp_sven)[1]))
colnames(data_temp_sven)[2] <- "Method"
data_temp <- rbind(data_temp_ns,data_temp_sven)

list_plots[["average"]] <- ggplot() +
  geom_line(data = data_temp, mapping = aes(x = Tenor,y = value, linetype = Model, color = Method), size = 1.2) +
  geom_point(data = data_temp_act, mapping = aes(x = Tenor,y = value, shape = variable), size = 2) +
  scale_color_manual(labels = str_labels, values = colours)+
  labs(y = "",x="Date",title="Average swap rate curve", shape = "") +
  guides(shape = guide_legend(order = 1),color = guide_legend(order = 3, ncol = 2), linetype = guide_legend(order = 2)) +
  theme_bw()+
  theme(legend.title=element_text(size = 17,face="plain",color="black"),
        legend.key.width = unit(1.2,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 18,face="plain",color="black"),
```

```
        axis.text = element_text(size = 18,face="plain",color="black"),
        legend.text = element_text(size = 15,face="plain",color="black"),
        plot.title = element_text(size = 19,face="plain",color="black"),
        #legend.position = c(0.68,0.43),
        legend.position = "none",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )

grid.arrange(arrangeGrob(list_plots[["average"]],list_plots[["2002-01"]]+theme(legend.position = c(0.68,0.43)),
                         list_plots[["2005-06"]],list_plots[["2008-08"]],
                         list_plots[["2011-10"]],list_plots[["2017-01"]],ncol=2))
```

A.6.2. Dynamics of the state variables

The following code includes: Figure 5.4, Figure 5.5, Figure 5.6

```
#####
## Dynamics of the state variables
# Nelson-Siegel
list_models_in_sample <- list_ts_state_variables_NS[1:4]
str_temp <- names(list_models_in_sample) <- c("2.step.fix","2.step.var","1.step","1.step.2.step")
state_var <- c("Level","Slope","Curvature")
data_temp <- NULL
date_vector <- index(xts_swap_rates)
n <- length(date_vector)
for (i in 1:3){
  for ( j in str_temp){
    data_temp <- rbind(data_temp,data.frame("Date" = date_vector,"StateVariable" = rep(state_var[i],n),"Method" = rep(j, n),"value" = as.numeric(list_models_in_sample[[j]][,i])))
  }
}
str_labels_plot <- c("2.step.fix","2.step.var","1.step","1.step.2.step")
colours <- blue2yellow(n = 5)[c(1,2,4)]
names(colours) <- state_var
str_labels <- c("2-step fix","2-step var","1-step", "1-2-step")
names(str_labels) <- str_labels_plot
ggplot(data = data_temp,aes(x = Date,y = value,linetype = Method, color = StateVariable))+
  geom_line(size=1.2) +
  labs(y = "",x="Date",title="")+
  scale_linetype_manual(labels = str_labels,values= c(1,4,2,5))+
  scale_color_manual(values = colours)+
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.key.width = unit(1.5,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = "right",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )

# Svensson model with 2 step var restricted
list_models_in_sample <- list_ts_state_variables_Sven[c(1,3:5)]
str_temp <- names(list_models_in_sample) <- c("2.step.fix","2.step.var","1.step","1.step.2.step")
state_var <- c("Level","Slope","Curvature1","Curvature2")
```

```
data_temp <- NULL
date_vector <- index(xts_swap_rates)
n <- length(date_vector)
for (i in 1:4){
  for ( j in str_temp){
    data_temp <- rbind(data_temp,data.frame("Date" = date_vector,"StateVariable" =
        rep(state_var[i],n),"Method" = rep(j, n),"value" = as.numeric(
        list_models_in_sample[[j]][,i])))
  }
}
str_labels_plot <- c("2.step.fix","2.step.var","1.step","1.step.2.step")
colours <- blue2yellow(n = 5)[c(1,2,4,5)]
names(colours) <- state_var
str_labels <- c("2-step fix","2-step var","1-step", "1-2-step")
names(str_labels) <- str_labels_plot

ggplot(data = data_temp,aes(x = Date,y = value,linetype = Method, color =
    StateVariable))+
  geom_line(size=1.2) +
  labs(y = "",x="Date",title="")+
  scale_linetype_manual(labels = str_labels,values= c(1,4,2,5))+
  scale_color_manual(values = colours)+
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.key.width = unit(1.5,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = "right",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  ) +
  guides(linetype = guide_legend(order=1),
         color = guide_legend(order=2))

# Svensson model with 2 step var unrestricted
list_models_in_sample <- list_ts_state_variables_Sven[c(1:2,4:5)]
str_temp <- names(list_models_in_sample) <- c("2.step.fix","2.step.var","1.step","1
    .step.2.step")
state_var <- c("Level","Slope","Curvature1","Curvature2")
data_temp <- NULL
date_vector <- index(xts_swap_rates)
n <- length(date_vector)
for (i in 1:4){
  for ( j in str_temp){
    data_temp <- rbind(data_temp,data.frame("Date" = date_vector,"StateVariable" =
        rep(state_var[i],n),"Method" = rep(j, n),"value" = as.numeric(
        list_models_in_sample[[j]][,i])))
  }
}
str_labels_plot <- c("2.step.fix","2.step.var","1.step","1.step.2.step")
colours <- blue2yellow(n = 5)[c(1,2,4,5)]
names(colours) <- state_var
str_labels <- c("2-step fix","2-step var unr","1-step", "1-2-step")
names(str_labels) <- str_labels_plot

ggplot(data = data_temp,aes(x = Date,y = value,linetype = Method, color =
    StateVariable))+
  geom_line(size=1.2) +
  labs(y = "",x="Date",title="")+
  scale_linetype_manual(labels = str_labels,values= c(1,4,2,5))+
  scale_color_manual(values = colours)+
```

```r
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.key.width = unit(1.5,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = "right",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  ) +
  guides(linetype = guide_legend(order=1),
         color = guide_legend(order=2))
```

A.6.3. Impulse response function

The following code includes: Figure 5.7

```r
#####
## IRF
#####
## Nelson Siegel
## 2-step fix
# State variables including Macroeconomic factors
xts_ns_macro <- cbind(list_ts_state_variables_NS$`2stepfix`[,c("Level","Slope","
    Curvature")],xts_macro[,c("Inflation_Eurozone","Outputgap_Eurozone", "
    EONIA_average")])
# estimate var
colnames(xts_ns_macro) <- c("Level","Slope","Curvature","Inflation","Outputgap","
    EONIA")
var_xts <- VAR(y = xts_ns_macro, p = 2, type = "const")

# irf function
# normal
state_variables <- c("Level","Slope","Curvature","Outputgap","Inflation","EONIA")
k <- 1; i <- "Level"; j <- "Slope"
runs_irf <- 100
stepsahead <- 59
var_temp <- var_xts
list_plots <- list()
for ( i in state_variables){
  for ( j in state_variables){
    irf_temp <- irf(x = var_temp,impulse = i, response = j, n.ahead = stepsahead,ci
        = 0.9,runs = runs_irf)
    data_temp <- data.frame("steps" = 1:(stepsahead+1)/12,
                            "irf" = irf_temp$irf[[i]],
                            "CI_L" = irf_temp$Lower[[i]],
                            "CI_U" = irf_temp$Upper[[i]])
    colnames(data_temp) <- c("steps","irf","CI_L","CI_U")
    data_temp <- melt(data_temp,"steps")
    y_lab = ""
    if (j == "Level"){
      y_lab = paste(i, "impulse")
    }
    x_lab = ""
    if ( i == "Level"){
      x_lab = paste(j, "response")
    }
    list_plots[[k]] <- ggplot(data = data_temp, mapping = aes(x = steps, y = value,
        color = variable, linetype = variable)) +
      geom_hline(yintercept = 0, linetype = 1, color = "black", size = 1) +
      geom_line(size = 1.2) +
```

```
      labs(y = y_lab, x = "", title = x_lab) +
      scale_linetype_manual(values=c(1,2,2))+
      scale_color_manual(values=c("blue","orange","orange"))+
      scale_x_continuous(limits = c(0,5), expand = c(0,0)) +
      ylim(-0.3,0.3) +
      theme_bw()+
      theme(legend.title=element_blank(),
            axis.title = element_text(size = 22,face="plain",color="black"),
            axis.text = element_text(size = 24,face="plain",color="black"),
            legend.text = element_text(size = 21,face="plain",color="black"),
            plot.title = element_text(size = 24,face="plain",color="black", hjust = 0.5),
            legend.position = "none",
            plot.margin = unit(c(0,0.2,0,0),"cm")
      )
    k <- k + 1
  }
}
plots_irf_const_2step <- list_plots

grid.arrange(arrangeGrob(
  plots_irf_const_2step[[1]],plots_irf_const_2step[[2]],plots_irf_const_2step[[3]],plots_irf_const_2step[[4]],plots_irf_const_2step[[5]],plots_irf_const_2step[[6]],plots_irf_const_2step[[7]],plots_irf_const_2step[[8]],plots_irf_const_2step[[9]],plots_irf_const_2step[[10]],plots_irf_const_2step[[11]],plots_irf_const_2step[[12]],
  plots_irf_const_2step[[13]],plots_irf_const_2step[[14]],plots_irf_const_2step[[15]],plots_irf_const_2step[[16]],plots_irf_const_2step[[17]],plots_irf_const_2step[[18]],plots_irf_const_2step[[19]],plots_irf_const_2step[[20]],plots_irf_const_2step[[21]],plots_irf_const_2step[[22]],plots_irf_const_2step[[23]],plots_irf_const_2step[[24]],
  plots_irf_const_2step[[25]],plots_irf_const_2step[[26]],plots_irf_const_2step[[27]],plots_irf_const_2step[[28]],plots_irf_const_2step[[29]],plots_irf_const_2step[[30]],plots_irf_const_2step[[31]],plots_irf_const_2step[[32]],plots_irf_const_2step[[33]],plots_irf_const_2step[[34]],plots_irf_const_2step[[35]],plots_irf_const_2step[[36]],
  ncol=6))
```

A.7. Out-of-sample forecasting

A.7.1. Forecast procedures - `1-step fix` and `1-2-step fix` preparation

```
##############################################
## Out of sample forecasting - one step forecasting preparation
## Forecasting one-step estimation methods
## time period for forecasting starting 2009-01 / 2014-09 (2nd regime)
fc_dates<-as.character(index(xts_swap_rates["2009-01/"]))
# 2nd regime: xts_swap_rates["2014-09/"]..
## SS presentations for Nelson-Siegel and Svensson model by dlm
# see estimation_methods_1step
#####
# dlm
SS_DLM_LL_1st_NS <- ... defined in estimation_methods_1step
SS_DLM_LL_1st_Sven <- ... defined in estimation_methods_1step
SS_DLM_NS <- ... defined in estimation_methods_1step
SS_DLM_Sven <- ... defined in estimation_methods_1step
# as well as lower/upper bounds defined in estimation_methods_1step

## Estimating the coefficients starting with 2009-01 / 2014-09 (2nd regime)
#####
# Nelson-Siegel
```

```
lambda0 <- as.numeric(list_ts_state_variables_NS$‘2stepfix‘[1,c("Lambda")])
NS_est_coef_fc_dlm <- list()
for ( i in fc_dates){
  start_time <- Sys.time()
  # date starting with the forecast
  fc_date_temp <- i
  # data for model estimation
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
  # 2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  # data preparation - initial values
  initial_values <- prep_1step_NS(lambda0, xts_swap_rates_temp)
  list_initial_values <- initial_values$initial_values
  P0 <- initial_values$P0
  x0 <- initial_values$x0
  # dlm
  NS_est_coef_fc_dlm[[fc_date_temp]]<-mle(SS_DLM_LL_1st_NS,start=list_initial_values
      ,method="L-BFGS-B",lower=lower_NS,upper = upper_NS)
  NS_est_coef_fc_dlm[[fc_date_temp]] <- coef(NS_est_coef_fc_dlm[[fc_date_temp]])
  # use information on lambda value as initial value
  lambda0 <- NS_est_coef_fc_dlm[[fc_date_temp]]["lambda"]
}

## Estimating the smoothed state variables
# smoothing by dlm package
lambda0 <- as.numeric(list_ts_state_variables_NS$‘2stepfix‘[1,c("Lambda")])
NS_est_smoothed_par_dlm_fc <- list()
for ( i in fc_dates){
  # date
  start_time <- Sys.time()
  fc_date_temp <- i
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
 # 2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  # data preparation
  initial_values <- prep_1step_NS(lambda0, xts_swap_rates_temp)
  P0 <- initial_values$P0
  x0 <- initial_values$x0
  # dlm
  SS_presentations_dlm_temp <- SS_DLM_NS(NS_est_coef_fc_dlm[[i]])

  mu0_yields <- matrix(t(FF(SS_presentations_dlm_temp) %*% NS_est_coef_fc_dlm[[i]][c
      ("m1","m2","m3")]),ncol=d_obs,nrow=n_obs,byrow=T)
  y_t_deflated = xts_swap_rates_temp - mu0_yields

  NS_est_smoothed_par_dlm_fc[[i]]<-dlmSmooth(y_t_deflated,SS_presentations_dlm_temp)
  NS_est_smoothed_par_dlm_fc[[i]]<-NS_est_smoothed_par_dlm_fc[[i]]$s[-1,]+matrix(
      NS_est_coef_fc_dlm[[i]][c("m1","m2","m3")],ncol=3,nrow=n_obs,byrow=T)
  NS_est_smoothed_par_dlm_fc[[i]]<-cbind(NS_est_smoothed_par_dlm_fc[[i]],rep(
      NS_est_coef_fc_dlm[[i]]["lambda"],n_obs))
  NS_est_smoothed_par_dlm_fc[[i]]<-xts(NS_est_smoothed_par_dlm_fc[[i]],index(
      xts_swap_rates_temp))
  colnames(NS_est_smoothed_par_dlm_fc[[i]])<-c("Level","Slope","Curvature","Lambda")
  # use information on lambda value as initial value
  lambda0 <- NS_est_coef_fc_dlm[[fc_date_temp]]["lambda"]
}

## dynamics of state variables by one-step-two-step estimation approach
NS_est_smoothed_par_dlm_12step_fc <- list()
df_swap_rates <- as.matrix(as.data.frame(xts_swap_rates))
# 2nd regime: ...xts_swap_rates["2012-08/",]))
for ( i in fc_dates){
  fc_date_temp <- i
```

```
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
  # 2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  lambda0 <- NS_est_coef_fc_dlm[[i]]["lambda"]
  X <- NULL
  for ( j in 1:n_obs){
    data_temp <- as.data.frame(cbind("curve" = df_swap_rates[j,],"tau" = tau_swap_rates))
    temp <- lm(curve ~ slope_loading(lambda0,tau)+curvature_loading(lambda0,tau), data=data_temp)
    X <- rbind(X,temp$coefficients)
  }
  NS_est_smoothed_par_dlm_12step_fc[[i]]<-cbind(X,rep(lambda0,n_obs))
  colnames(NS_est_smoothed_par_dlm_12step_fc[[i]])<-c("Level","Slope","Curvature","Lambda")
  NS_est_smoothed_par_dlm_12step_fc[[i]] <- xts(NS_est_smoothed_par_dlm_12step_fc[[i]],index(xts_swap_rates_temp))
}

#####
# Svensson
#####
lambda01<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("Lambda1")]) #(1Y)
lambda02<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("Lambda2")]) #(3Y)

Sven_est_coef_fc_dlm <- list()
for ( i in fc_dates){
  start_time <- Sys.time()
  # date starting with the forecast
  fc_date_temp <- i
  # data for model estimation
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
  #2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  # data preparation
  initial_values <- prep_1step_Sven(lambda01,lambda02, xts_swap_rates_temp)
  list_initial_values <- initial_values$initial_values
  P0 <- initial_values$P0
  x0 <- initial_values$x0
  # dlm
  Sven_est_coef_fc_dlm[[fc_date_temp]] <- mle(SS_DLM_LL_1st_Sven,start=list_initial_values,method="L-BFGS-B",lower = lower_Sven,upper = upper_Sven)
  Sven_est_coef_fc_dlm[[fc_date_temp]] <- coef(Sven_est_coef_fc_dlm[[fc_date_temp]])
  # use information on lambda valuea as initial values
  lambda01 <- Sven_est_coef_fc_dlm[[fc_date_temp]]["lambda1"]
  lambda02 <- Sven_est_coef_fc_dlm[[fc_date_temp]]["lambda2"]
}

## Estimating the smoothed state variables
# smoothing by dlm package
lambda01<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("lambda1")]) #(1Y)
lambda02<-as.numeric(list_ts_state_variables_Sven$`2stepfix`[1,c("lambda2")]) #(3Y)
Sven_est_smoothed_par_dlm_fc <- list()
for ( i in fc_dates){
  # date
  start_time <- Sys.time()
  fc_date_temp <- i
  # data for model estimation
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
 # 2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  # data preparation
  initial_values <- prep_1step_Sven(lambda01,lambda02, xts_swap_rates_temp)
```

```
  P0 <- initial_values$P0
  x0 <- initial_values$x0
 # dlm
  SS_presentations_dlm_temp <- SS_DLM_Sven(Sven_est_coef_fc_dlm[[i]])
  mu0_yields <- matrix(t(FF(SS_presentations_dlm_temp) %*% Sven_est_coef_fc_dlm[[i]][c("m1","m2","m3","m4")]), ncol = d_obs, nrow = n_obs, byrow = T)
  y_t_deflated = xts_swap_rates_temp - mu0_yields

  Sven_est_smoothed_par_dlm_fc[[i]]<-dlmSmooth(y_t_deflated, SS_presentations_dlm_temp)
  Sven_est_smoothed_par_dlm_fc[[i]] <- Sven_est_smoothed_par_dlm_fc[[i]]$s[-1,]+ matrix(Sven_est_coef_fc_dlm[[i]][c("m1","m2","m3","m4")],ncol=4,nrow=n_obs, byrow=T)
  Sven_est_smoothed_par_dlm_fc[[i]] <- cbind(Sven_est_smoothed_par_dlm_fc[[i]],rep(Sven_est_coef_fc_dlm[[i]]["lambda1"],n_obs),rep(Sven_est_coef_fc_dlm[[i]]["lambda2"],n_obs))
  Sven_est_smoothed_par_dlm_fc[[i]] <- xts(Sven_est_smoothed_par_dlm_fc[[i]], index(xts_swap_rates_temp))
  colnames(Sven_est_smoothed_par_dlm_fc[[i]]) <- c("Level","Slope","Curvature1","Curvature2","Lambda1","Lambda2")
  # use information on lambda value as initial values
  lambda01 <- Sven_est_coef_fc_dlm[[fc_date_temp]]["lambda1"]
  lambda02 <- Sven_est_coef_fc_dlm[[fc_date_temp]]["lambda2"]
}

## dynamics of state variables by one-step-two-step estimation approach
Sven_est_smoothed_par_dlm_12step_fc <- list()
df_swap_rates <- as.matrix(as.data.frame(xts_swap_rates))
# 2nd regime: ...xts_swap_rates["2012-08/",]))
for ( i in fc_dates){
  fc_date_temp <- i
  xts_swap_rates_temp <- xts_swap_rates[paste0("/",fc_date_temp)]
  #2nd regime: ...paste0("2012-08/",fc_date_temp)]
  n_obs <- dim(xts_swap_rates_temp)[1]
  lambda1 <- Sven_est_coef_fc_dlm[[i]]["lambda1"]
  lambda2 <- Sven_est_coef_fc_dlm[[i]]["lambda2"]
  X <- NULL
  for ( j in 1:n_obs){
    data_temp<-as.data.frame(cbind("curve"=df_swap_rates[j,],"tau"=tau_swap_rates))
    temp <- lm(curve ~ slope_loading(lambda1,tau)+curvature_loading(lambda1,tau)+curvature_loading(lambda2,tau), data = data_temp)
    X <- rbind(X,temp$coefficients)
  }
  Sven_est_smoothed_par_dlm_12step_fc[[i]]<-cbind(X,rep(lambda1,n_obs),rep(lambda2,n_obs))
  colnames(Sven_est_smoothed_par_dlm_12step_fc[[i]]) <- c("Level","Slope","Curvature1","Curvature2","Lambda1","Lambda2")
  Sven_est_smoothed_par_dlm_12step_fc[[i]]<-xts(Sven_est_smoothed_par_dlm_12step_fc[[i]],index(xts_swap_rates_temp))
}

#####
# evolution of the Lambda values
temp <- NULL
temp1 <- NULL
for ( i in names(NS_est_coef_fc_dlm)){
  temp <- rbind(temp,NS_est_coef_fc_dlm[[i]]["lambda"])
  temp1 <- rbind(temp1,Sven_est_coef_fc_dlm[[i]][c("lambda1","lambda2")])
}
NS_est_lambda <- xts(temp, as.Date(names(NS_est_coef_fc_dlm)))
colnames(NS_est_lambda) <- c("Lambda")
Sven_est_lambda <- xts(temp1, as.Date(names(NS_est_coef_fc_dlm)))
colnames(Sven_est_lambda) <- c("Lambda1","Lambda2")
```

A.7.2. Forecast procedures - VAR models

```
##############################################
## Out of sample forecasting
## VAR models forecasting state variables - swap rates - calculation residuals
##############################################
## Forecasting functions
## Out of sample Forecasting with endogenous macroeconomic factors
## forecast horizon: 1, 3, 6, 12, 24 months
## p .. lag of VAR specification
## type .. type of VAR specification "const" or "none" - including resp. excluding
    an intercept vector
## dt_start .. start of forecast
# function for methods: 2-step fix, 2-step var, 1-2-step
VAR_forecasting_out_sample_endogen_two_step <- function(sample, p, type, dt_start){
  h <- c(1,3,6,12,24) # forecast horizon
  h_max <- max(h)
  index <- as.character(index(sample))
  index.start <- which(dt_start == substr(index, 1,7))
  for ( i in index[index.start:(length(index)-1)]){
    xts.temp.forecasting <- as.matrix(sample[paste0("/",i)])
    # VAR model
    var.fitted <- VAR(xts.temp.forecasting, p ,type = type)
    phi <- Bcoef(var.fitted)
    alpha <- phi[,dim(phi)[2]]
    phi <- phi[,-dim(phi)[2]]
    # Forecasting over forecast horizon
    xts.temp.forecasting <- tail(xts.temp.forecasting,p)
    for ( k in 1:h_max){
      temp.forecast<-alpha+phi%*%as.numeric(t(tail(xts.temp.forecasting,p)[p:1,]))
      xts.temp.forecasting <- rbind(xts.temp.forecasting, t(temp.forecast))
    }
    if ( substr(i, 1,7) == dt_start ){
      forecast <- as.vector(t(xts.temp.forecasting[p+h,]))
    } else{
      forecast <- rbind(forecast, as.vector(t(xts.temp.forecasting[p+h,])))
    }
  }
  colnames(forecast) <- rep(colnames(sample), times =  length(h))
  list_forecast <- list()
  for ( i in 1:length(h)){
    temp <- dim(sample)[2]
    temp_index <- index[-(1:(index.start+h[i]-1))]
    list_forecast[[paste(h[i])]] <- xts(forecast[1:length(temp_index),((temp*i-temp
        +1):(temp*i))],as.Date(temp_index))
    # remove forecasts of the macroeconomic factors as not relevant for the interest
         rates forecasts
    n_col <- dim(list_forecast[[paste(h[i])]])[2]
    list_forecast[[paste(h[i])]] <- list_forecast[[paste(h[i])]][,-seq(n_col, by =
        -1, length = 3)]
  }
  list_forecast
}
# function for method: 1-step
VAR_forecasting_out_sample_endogen_one_step <- function(list_sample, p, type,
    str_params, dt_start){
  h <- c(1,3,6,12,24) # forecast horizon
  h_max <- max(h)
  for ( i in names(list_sample)){
    xts.temp.forecasting <- as.matrix(list_sample[[i]][,str_params])
    # VAR model
    var.fitted <- VAR(xts.temp.forecasting, p ,type = type)
    phi <- Bcoef(var.fitted)
```

```r
    alpha <- phi[,dim(phi)[2]]
    phi <- phi[,-dim(phi)[2]]
    # forecasting over forecast horizon
    xts.temp.forecasting <- tail(xts.temp.forecasting,p)
    for ( k in 1:h_max){
      temp.forecast<-alpha+phi%*%as.numeric(t(tail(xts.temp.forecasting,p)[p:1,]))
      xts.temp.forecasting <- rbind(xts.temp.forecasting, t(temp.forecast))
    }
    if ( substr(i, 1,7) == dt_start ){
      forecast <- as.vector(t(xts.temp.forecasting[p+h,]))
    } else{
      forecast <- rbind(forecast,as.vector(t(xts.temp.forecasting[p+h,])))
    }
  }
  colnames(forecast) <- rep(str_params, times =  length(h))
  list_forecast <- list()
  for ( i in 1:length(h)){
    temp <- length(str_params)
    temp_index <- fc_dates[-(1:(h[i]))]
    list_forecast[[paste(h[i])]] <- xts(forecast[1:length(temp_index),((temp*i-temp+1):(temp*i))],as.Date(temp_index))
    # remove forecasts of the macroeconomic factors not relevant for the interest rates forecasts
    n_col <- dim(list_forecast[[paste(h[i])]])[2]
    list_forecast[[paste(h[i])]]<-list_forecast[[paste(h[i])]][,-seq(n_col,by=-1,length=3)]
  }
  list_forecast
}

#####
## Out-of-sample forecasting of NS parameters with macro endogenous
## Total DataSample / Sub DataSample

type = "const"
list_forecast_out_sample <- list()
dt_start <- "2009-01"
# 2nd regime: dt_start <- "2014-09"
fc_dates <- index(xts_swap_rates["2009-01/"])
# 2nd regime: fc_dates <- index(xts_swap_rates["2014-09/"])
p <- 2
# two step estimation approaches with fixed decay parameters
str_methods  <- c("NS_2stepfix","Sven_2stepfix")
list_ts_state_variables_2step <- list("NS_2stepfix" = list_ts_state_variables_NS$`2stepfix`,"Sven_2stepfix" = list_ts_state_variables_Sven$`2stepfix`)
for ( i in str_methods){
  ## colums with lambda values
  if ( dim(list_ts_state_variables_2step[[i]])[2] == 4 ){
    n_col <- -4 # Nelson-Siegel
  } else {
    n_col <- c(-5,-6) # Svensson
  }
  sample <- merge(list_ts_state_variables_2step[[i]][,n_col],xts_all[,c("Inflation_Eurozone", "Outputgap_Eurozone", "EONIA_average")])
  # 2nd regime: ... variables_2step[[i]]["2012-08/",n_col],xts_all["2012-08/",c("Inflation_Eurozone", ...
  list_temp <- VAR_forecasting_out_sample_endogen_two_step(sample,p,type,dt_start)
  names(list_temp) <- paste(paste(paste(paste(i, paste("h", names(list_temp), sep="")), sep = "_"), "VAR", sep="_"),p,sep=""),"_Makro", sep="")
  # add the respective lambda values
  for ( j in names(list_temp)){
    temp <- xts(list_ts_state_variables_2step[[i]][fc_dates[1:dim(list_temp[[j]])[1]],-n_col],index(list_temp[[j]]))
```

```
    list_temp[[j]] <- merge(list_temp[[j]],temp)
  }
  list_forecast_out_sample <- c(list_forecast_out_sample,list_temp)
}
# two step estimation approaches with variable decay parameters
str_methods  <- c("NS_2stepvar","Sven_2stepvar")
list_ts_state_variables_2step <- list("NS_2stepvar" = list_ts_state_variables_NS$`2
    stepvar`,"Sven_2stepvar" = list_ts_state_variables_Sven$`2stepvar`)
for ( i in str_methods){
  ## colums with lambda values
  if ( dim(list_ts_state_variables_2step[[i]])[2] == 4 ){
    n_col <- -4 # Nelson-Siegel
  } else {
    n_col = c(-5,-6) # Svensson
  }
  sample <- merge(list_ts_state_variables_2step[[i]][,n_col],xts_all[,c("
      Inflation_Eurozone", "Outputgap_Eurozone", "EONIA_average")])
 # 2nd regime: ... variables_2step[[i]]["2012-08/",n_col],xts_all["2012-08/",c("
     Inflation_Eurozone", ...
  list_temp <- VAR_forecasting_out_sample_endogen_two_step(sample,p,type,dt_start)
  names(list_temp) <- paste(paste(paste(paste(i, paste("h", names(list_temp), sep=""
      ), sep = "_"), "VAR", sep="_"),p,sep=""),"_Makro", sep="")
  # add the respective lambda values
  list_temp_m_l <- list()
  for ( j in names(list_temp)){
    # last decay parameters
    temp <- xts(list_ts_state_variables_2step[[i]][fc_dates[1:dim(list_temp[[j]])
        [1]],-n_col],index(list_temp[[j]]))
    list_temp_m_l[[paste0(j,"_l")]] <- merge(list_temp[[j]],temp)
    # median of decay parameter
    n_row <- dim(list_temp[[j]])[1]
    list_temp_m_l[[paste0(j,"_m")]] <- NULL
    for ( k in 1:n_row){
      temp <- xts(t(apply(list_ts_state_variables_2step[[i]][paste0("/",fc_dates[k])
          ,-n_col],2,median)),index(list_temp[[j]])[k])
      # 2nd regime ... variables_2step[[i]][paste0("2012-08/",fc_dates[k]), ...
      list_temp_m_l[[paste0(j,"_m")]] <- rbind(list_temp_m_l[[paste0(j,"_m")]],merge
          (list_temp[[j]][k,],temp))
    }
    colnames(list_temp_m_l[[paste0(j,"_m")]]) <- colnames(list_temp_m_l[[paste0(j,"
        _l")]])
   }
  list_forecast_out_sample <- c(list_forecast_out_sample,list_temp_m_l)
}
# one step estimation approaches
str_methods  <- c("NS_1step","NS_12step","Sven_1step","Sven_12step")
list_ts_state_variables_1step <-
 list("NS_1step" = NS_est_smoothed_par_dlm_fc,
      # 2nd regime: NS_est_smoothed_par_dlm_fc_2nd_reg
      "NS_12step" = NS_est_smoothed_par_dlm_12step_fc,
      # 2nd regime: NS_est_smoothed_par_dlm_12step_fc_2nd_reg
      "Sven_1step" = Sven_est_smoothed_par_dlm_fc,
      # 2nd regime: Sven_est_smoothed_par_dlm_fc_2nd_reg
      "Sven_12step" = Sven_est_smoothed_par_dlm_12step_fc)
      #2nd regime: Sven_est_smoothed_par_dlm_12step_fc_2nd_reg
for ( i in str_methods){
  list_sample <- list_ts_state_variables_1step[[i]]
  ## colums with lambda values
  if ( dim(list_sample[[1]])[2] == 4 ){
    n_col <- -4 # Nelson-Siegel
  } else {
    n_col = c(-5,-6) # Svensson
  }
```

```
  for ( j in names(list_sample)){
    list_sample[[j]] <- merge(list_sample[[j]][,n_col],xts_all[paste0("/",j),c("
        Inflation_Eurozone", "Outputgap_Eurozone", "EONIA_average")])
   # 2nd regime: ...list_sample[[j]]["2012-08/",n_col],xts_all[paste0("2012-08/",j),
       c("Inflation_Eurozone",...
  }
  str_params_temp <- colnames(list_sample[[j]])
  list_temp <- VAR_forecasting_out_sample_endogen_one_step(list_sample,p,type,
      str_params_temp,dt_start)
  names(list_temp) <- paste(paste(paste(paste(i, paste("h", names(list_temp), sep=""
      ), sep = "_"), "VAR", sep="_"),p,sep=""),"_Makro", sep="")
  # add the respective lambda values
  temp <- NULL
  for ( j in names(list_ts_state_variables_1step[[i]])){
    temp <- rbind(temp,list_ts_state_variables_1step[[i]][[j]][1,-n_col])
  }
  for ( j in names(list_temp)){
    n_row = dim(list_temp[[j]])[1]
    list_temp[[j]]<-merge(list_temp[[j]],xts(temp[1:n_row,],index(list_temp[[j]])))
  }
  list_forecast_out_sample <- c(list_forecast_out_sample,list_temp)
}
## Computation of Swap Rates Forecasts and Residuals to actual Swap Rates
list_forecast_out_sample_residuals <- list()
for ( i in names(list_forecast_out_sample)){
  # calculate swap rates and residuals
  list_forecast_out_sample_residuals[[i]] <- NULL
  if ( grepl("NS", i)){ # Nelson-Siegel
    for ( j in as.character(index(list_forecast_out_sample[[i]]))){
      list_forecast_out_sample_residuals[[i]]<-rbind(
          list_forecast_out_sample_residuals[[i]],nelson_siegel_model(as.numeric(
          list_forecast_out_sample[[i]][j,]),tau_swap_rates))
    }
  } else { # Svensson
    for ( j in as.character(index(list_forecast_out_sample[[i]]))){
      list_forecast_out_sample_residuals[[i]] <- rbind(
          list_forecast_out_sample_residuals[[i]],svensson_model(as.numeric(
          list_forecast_out_sample[[i]][j,]),tau_swap_rates))
    }
  }
  list_forecast_out_sample_residuals[[i]] <- xts(list_forecast_out_sample_residuals
      [[i]], index(list_forecast_out_sample[[i]]))
  list_forecast_out_sample_residuals[[i]] <- list_forecast_out_sample_residuals[[i]]
     - xts_swap_rates[index(list_forecast_out_sample[[i]]),]
}
```

A.7.3. Out-of-sample results

The following code includes: Table 6.1, Figure 6.1, Figure 6.2, Figure 6.3, Table 6.2, Figure 6.4

```
###########
## Out-of-sample forecast results
#####
## Total Datasample
str_models <- NULL
for (i in c("h1_","h3_","h6_","h12","24")){
  for ( j in c("NS_2stepfix","NS_2stepvar","NS_1step","NS_12step","Sven_2stepfix","
      Sven_2stepvar","Sven_1step","Sven_12step")){
    str_models <- c(str_models,names(list_forecast_out_sample_residuals)[grepl(i,
        names(list_forecast_out_sample_residuals)) & grepl(j, names(
        list_forecast_out_sample_residuals))])
```

```
  }
}
## Out-of-sample term-structure curve forecasts
table_termstructure(list_forecast_out_sample_residuals, str_models)
# [1] "2009-02-28"
# [1] "2017-08-31"
#                                  Mean Median Std. Dev.   Min.   Max.
# NS_2stepfix_h1_VAR2_Makro      0.1741 0.1558    0.0772 0.0690 0.4996
# NS_2stepvar_h1_VAR2_Makro_l    0.1763 0.1608    0.0888 0.0512 0.5095
# NS_2stepvar_h1_VAR2_Makro_m    0.2718 0.2399    0.1531 0.0788 0.7369
# NS_1step_h1_VAR2_Makro         0.1819 0.1629    0.0857 0.0557 0.5593
# NS_12step_h1_VAR2_Makro        0.1709 0.1508    0.0794 0.0620 0.4998
# Sven_2stepfix_h1_VAR2_Makro    0.1639 0.1446    0.0778 0.0624 0.4921
# ....
## Dynamics of the estimated decay parameters
fc_dates <- as.character(index(xts_swap_rates["2009-01/"]))
lambdas <- cbind(fc_dates,as.data.frame(merge(NS_est_lambda,
    list_ts_state_variables_NS$`2stepvar`[fc_dates,"Lambda"],Sven_est_lambda,
    list_ts_state_variables_Sven$`2stepvar`[fc_dates,c("Lambda1","Lambda2")])))
## Nelson-Siegel
data_temp <- rbind(cbind(melt(data.frame("Date"=fc_dates,"Lambda" = as.data.frame(
    NS_est_lambda)[,1]),"Date"), "Method" = rep("1.step",times = length(fc_dates))),
    cbind(melt(data.frame("Date"=fc_dates,as.data.frame(list_ts_state_variables_NS$
    `2stepvar`[fc_dates,"Lambda"])),"Date"), "Method" = rep("2.step",times = length(
    fc_dates))))

str_lab <- expression(lambda)
names(str_lab) <- "Lambda"
colours <- blue2yellow(n = 3)[1]
str_lab_method <- c("1-step","2-step var")
names(str_lab_method) <- c("1.step","2.step")

plot_ns_both <- ggplot(data = data_temp,aes(x = Date,y = value, color = variable,
    linetype = Method))+
  geom_line(size=1.2)+
  labs(y = "",x="Date",title="Nelson-Siegel")+
  scale_linetype_manual(labels = str_lab_method,values=c(1,2))+
  scale_color_manual(labels = str_lab,values = colours)+
  theme_bw()+
  theme(legend.title=element_blank(),
        legend.key.width = unit(1.5,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        legend.text = element_text(size = 20,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = c(0.8,0.8),
        plot.margin = unit(c(0,0.7,0,0),"cm")
  )
## Svensson
data_temp <- rbind(cbind(melt(data.frame("Date"=fc_dates,as.data.frame(
    Sven_est_lambda)),"Date"), "Method" = rep("1.step",times = length(fc_dates))),
    cbind(melt(data.frame("Date"=fc_dates,as.data.frame(list_ts_state_variables_Sven
    $`2stepvar`[fc_dates,c("Lambda1","Lambda2")])),"Date"), "Method" = rep("2.step",
    times = length(fc_dates))))

str_lab <- expression(lambda^1, lambda^2)
names(str_lab) <- c("Lambda1","Lambda2")
colours <- blue2yellow(n = 5)[c(1,3)]
str_lab_method <- c("1-step","2-step var")
names(str_lab_method) <- c("1.step","2.step")

```

```
ggplot(data = data_temp,aes(x = Date,y = value, color = variable, linetype = Method)
    )+
  geom_line(size=1.2)+
  labs(y = "",x="Date",title="Svensson")+
  scale_linetype_manual(labels = str_lab_method,values=c(1,2))+
  scale_color_manual(labels = str_lab,values = colours)+
  theme_bw()+
  theme(legend.title=element_blank(),
        legend.key.width = unit(1.5,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 22,face="plain",color="black"),
        legend.text = element_text(size = 20,face="plain",color="black"),
        plot.title = element_text(size = 24,face="plain",color="black"),
        legend.position = c(0.8,0.6),
        plot.margin = unit(c(0,0.7,0,0),"cm")
  )

#######
## Out-of sample forecasts graphs
str_temp <- str_models[grepl("h12",str_models)] # and str_temp <- str_models[grepl("
    h1_",str_models)]

## Out-of sample forecasts over tenor
data_temp <- table_tenor(list_forecast_out_sample_residuals,str_temp)$Mean
data_temp <- cbind("Tenor" = tau_swap_rates,as.data.frame(data_temp))
# Nelson-Siegel
data_temp_fit_ns <- data_temp[,c(1,2,3,4,5,6)]
colnames(data_temp_fit_ns)[-1] <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","
    X1.step","X1.step.2.step")
data_temp_fit_ns <- melt(data_temp_fit_ns, "Tenor")
data_temp_fit_ns <- cbind(data_temp_fit_ns,"Model" = rep("Nelson-Siegel",dim(
    data_temp_fit_ns)[1]))
colnames(data_temp_fit_ns)[2] <- "Method"
# Svensson
data_temp_fit_sven <- data_temp[,c(1,7,8,9,10,11)]
colnames(data_temp_fit_sven)[-1] <- c("X2.step.fix","X2.step.var.l","X2.step.var.m",
    "X1.step","X1.step.2.step")
data_temp_fit_sven <- melt(data_temp_fit_sven, "Tenor")
data_temp_fit_sven <- cbind(data_temp_fit_sven,"Model" = rep("Svensson",dim(
    data_temp_fit_sven)[1]))
colnames(data_temp_fit_sven)[2] <- "Method"
data_temp <- rbind(data_temp_fit_ns,data_temp_fit_sven)

## Beschriftung der modelle in Legende verallgemeinert.
str_labels_plot <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","X1.step","
    X1.step.2.step")
colours <- blue2yellow(n = 5)
names(colours) <- str_labels_plot
str_labels <- c("2-step fix", "2-step var-l","2-step var-m","1-step","1-2-step")
names(str_labels) <- str_labels_plot

ggplot(data = data_temp,aes(x = Tenor,y = value,linetype = Model, col = Method))+
  geom_line(size=1.2)+
  labs(y = "",x="Tenor",title="One-month forecast horizon")+
  #  scale_linetype_manual(labels = str_models_temp,values=c(1,2,4,5,6,7))+
  scale_color_manual(labels = str_labels, values = colours)+
  guides(color = guide_legend(order = 2), linetype = guide_legend(order = 1)) +
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.spacing.x = unit(0.1,"cm"),
        legend.margin = margin(0, 0, 0, 0, "cm"),
        legend.key.width = unit(1,"cm"),
```

```
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 21,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 20,face="plain",color="black"),
        #legend.position = c(0.8,0.3),
        legend.position = "none",
        plot.margin = unit(c(0,0.1,0,0),"cm")
  )
Mean_over_tenor_h1 <- Mean_over_tenor
#Mean_over_tenor_h12 <- Mean_over_tenor
## Combine two mean over tenor plots
combined <- Mean_over_tenor_h1 + Mean_over_tenor_h12 & theme(legend.position = "bottom")
combined <- combined + plot_layout(guides = "collect")

## Out-of sample forecasts over time - twelve month forecast horizon
rmse_swapratecurve <- NULL
for ( j in str_temp){
  rmse_swapratecurve <- cbind(rmse_swapratecurve,apply(list_forecast_out_sample_residuals[[j]],1,RMSE_fun))
}
data_temp <- cbind("Date" = index(list_forecast_out_sample_residuals[[str_temp[1]]]),as.data.frame(rmse_swapratecurve))
colnames(data_temp)[-1] <- c("X2.step.fix.NS","X2.step.var.l.NS","X2.step.var.m.NS","X1.step.NS","X1.step.2.step.NS","X2.step.fix.Sven","X2.step.var.l.Sven","X2.step.var.m.Sven","X1.step.Sven","X1.step.2.step.Sven")
# Nelson-Siegel
data_temp_fit_ns <- data_temp[,c(1,2,3,4,5,6)]
colnames(data_temp_fit_ns)[-1] <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","X1.step","X1.step.2.step")
data_temp_fit_ns <- melt(data_temp_fit_ns, "Date")
data_temp_fit_ns <- cbind(data_temp_fit_ns,"Model" = rep("Nelson-Siegel",dim(data_temp_fit_ns)[1]))
colnames(data_temp_fit_ns)[2] <- "Method"
# Svensson
data_temp_fit_sven <- data_temp[,c(1,7,8,9,10,11)]
colnames(data_temp_fit_sven)[-1] <- c("X2.step.fix","X2.step.var.m","X2.step.var.l","X1.step","X1.step.2.step")
data_temp_fit_sven <- melt(data_temp_fit_sven, "Date")
data_temp_fit_sven <- cbind(data_temp_fit_sven,"Model" = rep("Svensson",dim(data_temp_fit_sven)[1]))
colnames(data_temp_fit_sven)[2] <- "Method"
data_temp <- rbind(data_temp_fit_ns,data_temp_fit_sven)

## Beschriftung der modelle in Legende verallgemeinert.
str_labels_plot <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","X1.step","X1.step.2.step")
colours <- blue2yellow(n = 5)
names(colours) <- str_labels_plot
str_labels <- c("2-step fix","2-step var-l", "2-step var-m","1-step","1-2-step")
names(str_labels) <- str_labels_plot

ggplot(data = data_temp,aes(x = Date,y = value,linetype = Model, col = Method))+
  geom_line(size=1.2)+
  labs(y = "RMSE",x="Date",title="Twelve-month forecast horizon")+
  #  scale_linetype_manual(labels = str_models_temp,values=c(1,2,4,5,6,7))+
  scale_color_manual(labels = str_labels, values = colours)+
  guides(color = guide_legend(order = 2), linetype = guide_legend(order = 1)) +
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.spacing.x = unit(0.1,"cm"),
        legend.margin = margin(0, 0, 0, 0, "cm"),
```

```
        legend.key.width = unit(1,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 21,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 20,face="plain",color="black"),
        legend.position = c(0.9,0.72),
        #legend.position = "none",
        plot.margin = unit(c(0,0.6,0,0),"cm")

## Sub Datasample
str_models <- NULL
for (i in c("h1_","h3_","h6_","h12")){
  for ( j in c("NS_2stepfix","NS_2stepvar","NS_1step","NS_12step","Sven_2stepfix","
      Sven_2stepvar","Sven_1step","Sven_12step")){
    str_models <- c(str_models,names(list_forecast_out_sample_2nd_reg_residuals)[
        grepl(i, names(list_forecast_out_sample_2nd_reg_residuals)) & grepl(j, names
        (list_forecast_out_sample_2nd_reg_residuals))])
  }
}
## Out-of-sample term-structure curve forecasts
table_termstructure(list_forecast_out_sample_2nd_reg_residuals, str_models)
# [1] "2014-10-31"
# [1] "2017-08-31"
#                                  Mean Median Std. Dev.   Min.    Max.
# NS_2stepfix_h1_VAR2_Makro       0.1546 0.1542    0.0689 0.0717  0.3354
# NS_2stepvar_h1_VAR2_Makro_l     0.1406 0.1204    0.0729 0.0519  0.3036
# NS_2stepvar_h1_VAR2_Makro_m     0.1738 0.1449    0.0911 0.0797  0.3715
# NS_1step_h1_VAR2_Makro          0.1548 0.1425    0.0709 0.0583  0.3762
# NS_12step_h1_VAR2_Makro         0.1468 0.1418    0.0733 0.0536  0.3271
# Sven_2stepfix_h1_VAR2_Makro     0.1551 0.1349    0.0899 0.0548  0.4526
# ...

#######
## Out-of sample forecasts graphs
str_temp <- str_models[grepl("h12",str_models)] # and str_temp <- str_models[grepl("
    h1_",str_models)]

# Out-of sample forecasts over time - twelve month forecast horizon
#####
rmse_swapratecurve <- NULL
for ( j in str_temp){
  rmse_swapratecurve <- cbind(rmse_swapratecurve,apply(
      list_forecast_out_sample_2nd_reg_residuals[[j]],1,RMSE_fun))
}
data_temp <- cbind("Date" = index(list_forecast_out_sample_2nd_reg_residuals[[
    str_temp[1]]]),as.data.frame(rmse_swapratecurve))
colnames(data_temp)[-1] <- c("X2.step.fix.NS","X2.step.var.l.NS","X2.step.var.m.NS",
    "X1.step.NS","X1.step.2.step.NS","X2.step.fix.Sven","X2.step.var.l.Sven","
    X2.step.var.m.Sven","X1.step.Sven","X1.step.2.step.Sven")
# Nelson-Siegel
data_temp_fit_ns <- data_temp[,c(1,2,3,4,5,6)]
colnames(data_temp_fit_ns)[-1] <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","
    X1.step","X1.step.2.step")
data_temp_fit_ns <- melt(data_temp_fit_ns, "Date")
data_temp_fit_ns <- cbind(data_temp_fit_ns,"Model" = rep("Nelson-Siegel",dim(
    data_temp_fit_ns)[1]))
colnames(data_temp_fit_ns)[2] <- "Method"
# Svensson
data_temp_fit_sven <- data_temp[,c(1,7,8,9,10,11)]
colnames(data_temp_fit_sven)[-1] <- c("X2.step.fix","X2.step.var.m","X2.step.var.l",
    "X1.step","X1.step.2.step")
data_temp_fit_sven <- melt(data_temp_fit_sven, "Date")
```

```
data_temp_fit_sven <- cbind(data_temp_fit_sven,"Model" = rep("Svensson",dim(
    data_temp_fit_sven)[1]))
colnames(data_temp_fit_sven)[2] <- "Method"
data_temp <- rbind(data_temp_fit_ns,data_temp_fit_sven)

## Beschriftung der modelle in Legende verallgemeinert.
str_labels_plot <- c("X2.step.fix","X2.step.var.l","X2.step.var.m","X1.step","
    X1.step.2.step")
colours <- blue2yellow(n = 5)
names(colours) <- str_labels_plot
str_labels <- c("2-step fix","2-step var-l", "2-step var-m","1-step","1-2-step")
names(str_labels) <- str_labels_plot

ggplot(data = data_temp,aes(x = Date,y = value,linetype = Model, col = Method))+
  geom_line(size=1.2)+
  labs(y = "RMSE",x="Date",title="Twelve-month forecast horizon")+
  #  scale_linetype_manual(labels = str_models_temp,values=c(1,2,4,5,6,7))+
  scale_color_manual(labels = str_labels, values = colours)+
  guides(color = guide_legend(order = 2), linetype = guide_legend(order = 1)) +
  theme_bw()+
  theme(legend.title=element_text(size = 22,face="plain",color="black"),
        legend.spacing.x = unit(0.1,"cm"),
        legend.margin = margin(0, 0, 0, 0, "cm"),
        legend.key.width = unit(1,"cm"),
        legend.text.align = 0,
        axis.title = element_text(size = 22,face="plain",color="black"),
        axis.text = element_text(size = 21,face="plain",color="black"),
        legend.text = element_text(size = 21,face="plain",color="black"),
        plot.title = element_text(size = 20,face="plain",color="black"),
        legend.position = c(0.9,0.72),
        #legend.position = "none",
        plot.margin = unit(c(0,0.6,0,0),"cm")
```