Erasmus University Rotterdam

Erasmus School of Economics

MSc Econometrics and Management Science - Quantitative Finance

October 29, 2023

Proximal-Parameter Updates in a Score-Driven Dynamic Nelson-Siegel Framework

Supervisor

Author   Dr. Rutger-Jan Lange

Pieter van der Vleuten   482012   *Second Assessor*

Prof. dr. Michel van der Wel

Abstract

The goal of this thesis is to fabricate a yield curve forecasting tool that beats widely used benchmark methods like the dynamic conditional score filter and the Kalman filter. To this end, we propose a novelty to the literature by applying the proximal-parameter (ProPar) filter of Lange et al. (2022) to the dynamic Nelson-Siegel framework. The framework estimates and forecasts yield curves via time-varying level, slope, and curvature factors. ProPar updates the factors by maximizing the log-likelihood contribution of the current observation while penalizing around the one-step-ahead prediction. Simulations and empirical analysis on U.S. yield data demonstrate ProPar's out-of-sample forecasting accuracy and superiority over other widely used filtering methods. While the risk of overfitting is readily apparent when dealing with yield data, the outcomes of proximal-parameter updates are encouraging.

Keywords: proximal-parameter updates, Nelson-Siegel model, implicit gradient, yield forecasting, dynamic conditional score filter, Kalman filter, GAS-Kalman filter, time-varying parameter models

Note: The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Contents

1 Introduction

Grasping the dynamic development of the term structure is important in many areas of finance, including managing financial risk, pricing financial assets, allocating portfolios, valuing capital goods, and conducting monetary policy. Being able to model the yield curve allows one to make predictions. This thesis combines the proximal-parameter (ProPar) filtering method introduced by Lange et al. (2022) and the popular dynamic Nelson and Siegel (1987) framework, intending to fabricate a yield curve forecasting tool that beats the popular benchmark models.

The Nelson-Siegel framework is a popular way to model the yield curve by its level, slope, and curvature factor via a parametric function. Diebold and Li (2006) establish that the dynamic version, in which the factors are treated as dynamic processes, is appropriate for yield curve forecasting. The main challenge in the dynamic Nelson-Siegel framework is how to accurately predict the time-varying factors that are used to construct yield curve forecasts. This is where the ProPar filtering method comes into play. The ProPar filter updates the factors by maximizing the log-likelihood contribution of the current observation while adding a penalty centered around the one-step-ahead prediction. This setup implies an implicit stochastic-gradient update and has several advantages over its more commonly used explicit counterpart (Lange et al., 2022).

This study is the first to use proximal parameter updates in a dynamic Nelson-Siegel framework. Numerous filtering methods have already been applied to the framework, among others the popular Kalman filter (Koopman et al., 2010) and the dynamic conditional score (DCS) filter (Quaedvlieg and Schotman, 2022). Considering the promising findings on proximal parameter updates of Lange et al. (2022), we aim to fabricate a yield forecasting tool that resembles the application op the ProPar filter to the Nelson-Siegel framework.

This brings us to the research questions:

1. Is the score-driven dynamic Nelson-Siegel model with proximal-parameter updates a suitable framework for forecasting the term structure of interest rates?

2. Is the ProPar filter in a Nelson-Siegel framework able to outperform several widely-used benchmark models, among which its explicit counterpart and the popular Kalman filter?

To evaluate the ProPar filtering method and several benchmark methods in a dynamic Nelson-Siegel framework, this thesis forecasts U.S. yields. A data set containing monthly U.S. yields derived from government bonds is provided by Liu and Wu (2021). We utilize data from a period of 50 years ranging from November 1971 to October 2021, and 16 different maturities ranging from 3 to 180 months.

We distinguish two variants of the dynamic Nelson-Siegel model. First, we discuss a model in which only the level, slope, and curvature factors are considered to be time-varying. We refer to this model as the NS3F model. Second, we additionally treat the shape parameter that governs the shape of the factor loading matrix as time-varying. We call this model the NS4F model. The time-varying factors in both models are predicted using the ProPar filter. To answer the research question, we make a direct comparison with three benchmark filtering methods: the dynamic conditional score (DCS) filter, which is the explicit equivalent of the ProPar filter; the widely used Kalman filter; and the self-devised GAS-Kalman filter, resembling the Kalman filter but utilizing an autoregressive score specification for the factor dynamics. Next to these advanced filtering methods, we also compare the results with rather straightforward models such as the first-order vector autoregressive (VAR[1]) model and a naive forecasting model.

We conduct a simulation experiment involving the generation of yield data with four different specifications. These specifications encompass the three- and four-factor Nelson-Siegel frameworks, utilizing both a generalized autoregressive score (GAS) model and a state-space model for modeling the dynamic process of the factors. The results from the simulation demonstrated that the ProPar filter, along with the considered benchmark filtering methods, exhibits the capability to fit the data and provide out-of-sample yield forecasts that closely approximate the "true model", i.e. the model that uses the true parameter values. Notably, the Kalman filter outperforms the benchmark methods in terms of out-of-sample forecasting within this simulation scenario.

However, the findings from the empirical study using U.S. yield data reveals a different perspective. In this context, the ProPar filtering method surpasses its benchmark filtering methods, albeit by a narrow margin, in terms of out-of-sample yield forecasting. Unexpectedly, the naive forecasting model beats the advanced ProPar filter in terms of out-of-sample forecasting. This result confirms the persistence in the factors, and warns for the problem of overfitting.

The results show that the ProPar filter is an appropriate method for yield forecasting via the Nelson-Siegel framework. It can predict the one-step-ahead yield curves at least as well as its explicit counterpart, which is the dynamic conditional score filter, or the popular Kalman filter. Although the danger of overfitting is clearly present when working with yield data, this thesis provides a promising foundation for proximal parameter updates in a Nelson-Siegel framework.

The remainder of this thesis is structured as follows. In the next part of this section, Section 1.1, we discuss relevant literature on yield forecasting and the ProPar filter. Section 2 gives a detailed introduction to the Nelson-Siegel framework, while Section 3 walks through the steps of the ProPar filtering method. The simulation experiment in Section 4 validates the use of the ProPar filtering method in the Nelson-Siegel framework, whereas the empirical study in Section 5 applies the concept to real-world data. Finally, Section 6 answers the research questions and offers concluding remarks.

1.1 Literature Review

Precisely modeling the yield curve enables accurate predictions regarding future outcomes. For instance, according to Harvey (1989), the term structure of interest rates can be used to forecast economic growth. Harvey (1991a), Harvey (1991b), Harvey (1991c), Harvey et al. (1992), Harvey (1993), and Morrison (1992) conclude that a simple interest-rate-based model works in almost all of the G-7 countries, while Alvarez et al. (1992) apply the model effectively in an emerging country. Discrepancies in the yield curves of countries can potentially anticipate variations in economic growth. An implication of economic growth generally leads to the desire to forecast financial stock returns. This can also be accomplished by looking at the term structure (Campbell, 1987; Fama and French, 1989).

As one would anticipate, much literature has been devoted to modeling yield curves. However, methods with the purpose of yield curve forecasting have received relatively limited attention. When it comes to yield curve modeling, no-arbitrage models and equilibrium models emerge as the main techniques. The class of arbitrage-free models, elaborated on by Hull and White (1990) and Heath et al. (1992), is not appropriate for forecasting as its primary focus is to achieve a precise fit of the yield curve cross-sectionally. On the other hand, equilibrium models (among others Vasicek (1977) and Cox et al. (1985)) are based on dynamics influenced by the short rate, making them potentially relevant to forecasting. However, its literature concentrates on obtaining an accurate fit within the sample data, rather than out-of-sample forecasts. Besides, studies that do use affine equilibrium models to forecast out-of-sample consistently conclude that the models perform inadequately (Duffee, 2002; Egorov et al., 2006).

Alternatively, the class of Nelson-Siegel models, stemming from Nelson and Siegel (1987), enjoys immense popularity in practical applications, captivating the interest of financial market practitioners and central banks alike (Svensson, 1994; BIS, 2005). Nelson and Siegel (1987) propose a flexible, smooth parametric function that aims to fit the term structure. Their model can capture many of the typically observed shapes that the yield curve assumes over time. The base model consists of three factors that represent the level, slope, and curvature of the yield curve and thus convey some level of economic interpretation. The prominent study of Diebold and Li (2006) comes up with a dynamic version of the Nelson-Siegel model that produces encouraging forecasting results. It treats the level, slope, and curvature factors as three time-varying processes. They estimate the dynamic aspect of the factors using a vector autoregressive (VAR) process and use the predicted factors to construct yield curve forecasts.

Diebold and Li (2006) decide to fix the shape parameter λ_t, which determines the shape of the factor loadings, at a prespecified value to enhance simplicity, convenience, and numerical trustworthiness. Fixing λ_t ensures a linear setting for the estimation of the time-varying factors. Koopman et al. (2010) contribute to the literature by treating the shape parameter λ_t as a fourth latent factor, resulting in a nonlinear dynamic model. Also, they introduce time-varying volatility, where Diebold and Li (2006) assume a constant variance for each maturity and for the full sample. The most important finding of Koopman et al. (2010) is that a time-varying shape parameter and a time-varying volatility both lead to significant improvements in the model fit.

Finding accurate forecasts of the Nelson-Siegel factors is key to generating accurate yield curve forecasts. Diebold and Li (2006) use cross-section least squares to estimate the three factors at each point in time. Subsequently, they model the dynamics of these estimated factors via a rather simple VAR(1) process. Diebold et al. (2006) introduce a state-space formulation of the Nelson-Siegel model, treating the three factors as unobserved processes and modeling them using VAR processes. Koopman et al. (2010) adopt this approach, with state-space representation of the Nelson-Siegel model as a fundamental component. They consider the linear problem with a fixed shape parameter as a linear Gaussian state-space model, which they estimate via the Kalman filter (Kalman, 1960). As the Kalman filter method is only applicable to models that possess linearity in the state vector and the extensions that are considered by Koopman et al. (2010) introduce nonlinearities, they locally linearize the observation equation. This leads to the extended Kalman filter (Anderson and Moore, 2012). As the linearizations in the observation equation may lead to losses in accuracy, Quaedvlieg and Schotman (2022) model the time-varying shape parameter using a dynamic conditional score (DCS) specification. DCS is an observation-driven modeling principle, which proposes to update parameters in the direction of their likelihood gradient, i.e. the score function.

Lange et al. (2022) introduce the proximal-parameter (ProPar) filter method, which updates parameters by maximizing the observation log-density with respect to the parameter vector while penalizing the weighted, squared Euclidean norm relative to the one-step-ahead prediction. This implies an implicit stochastic-gradient update. According to Lange et al. (2022), the ProPar filter method broadens several attractive properties of explicit score-driven models (e.g. the DCS specification used by Quaedvlieg and Schotman (2022)) from the local to the global setting. Moreover, it possesses stronger contraction properties, resulting in an effective filter notwithstanding several types of misspecification. This thesis will model the dynamic factors of the dynamic Nelson-Siegel framework using the proximal-parameter (ProPar) filter method. Not only will we see whether the combination of this rather new filtering method and the Nelson-Siegel framework provides a good yield forecasting tool, but also what the dynamic estimation and forecasting performances of the ProPar filter method are compared to its explicit variant.

2 Nelson-Siegel framework

To examine the effect of the ProPar filter within the Nelson-Siegel framework, we initially consider the level, slope, and curvature factors as the only time-varying factors. Section 2.1 presents the resulting observation equation and the dynamic process of the three factors. Later on, in Section 2.2, we extend the model with time-varying shape parameter λ_t.

2.1 Dynamic Nelson-Siegel model

The observation equation of the Nelson-Siegel three-factor base model, from here on referred to as the NS3F model, is given by

$$y_t = \Lambda(\lambda)\beta_t + \epsilon_t, \qquad \epsilon_t \sim \text{NID}(0, \Sigma_\epsilon), \tag{1}$$

where $y_t = (y_t(\tau_1), ..., y_t(\tau_N))'$ is the vector containing N yields at time t, β_t contains the $K = 3$ time-varying (level, slope, and curvature) factors, and $\epsilon_t = (\epsilon_{1t}, ..., \epsilon_{Nt})'$ is the disturbance vector. $\Lambda(\lambda)$ is the $N \times 3$ factor loading matrix which elements depend on shape parameter $\lambda > 0$ and for which the $(i, j)^{\text{th}}$ element is given by

$$\Lambda_{ij}(\lambda) = \begin{cases} 1, & j = 1 \\ (1 - e^{-\lambda\tau_i})/\lambda\tau_i & j = 2 \\ (1 - e^{-\lambda\tau_i} - \lambda\tau_i e^{-\lambda\tau_i})/\lambda\tau_i & j = 3. \end{cases} \tag{2}$$

In the NS3F model where shape parameter λ is assumed to be constant over time, $\Lambda(\lambda)$ will be denoted as Λ for convenience.

The dynamics of β_t can be described by any dynamic model. Similar to Diebold and Li (2006), we use a rather simple first-order vector autoregressive (VAR[1]) model given by

$$\beta_{t+1} = \omega + \Phi\beta_t + \eta_{t+1}, \qquad \eta_t \sim \text{NID}(0, \Sigma_\eta), \tag{3}$$

in which ω is a $K \times 1$ vector of constants, Φ the $K \times K$ autoregressive coefficient matrix, and η_t the $K \times 1$ disturbance vector zero zero mean and variance Σ_η.

It is important to note that the level, slope and curvature factors do not literally represent the level, slope and curvature of the yield curve. Instead, the level factor equals the long-term yield ($y(\infty)$), as the factor loadings are 1 for the level factor and 0 for the slope and curvature factors when the maturity is ∞. It is fair to assume that the long-term yield, and hence the level factor, are positive from an economic perspective. The slope factor can be explained as the instant yield minus the long-term yield ($y(0) - y(\infty)$) as the sum of factor loadings in this construction is 1 for the slope factor and 0 for the level and curvature factor. As yield curves are generally upward sloping, implying $y(0) < y(\infty)$, the slope factor is negative on average. Thus, it is important to distinguish the slope and the slope factor of the yield curve, as they tend to have a different sign. The curvature factor is highly correlated to more complex portfolio of yields. Diebold and Li (2006) define the curvature factor as twice the two-year yield minus the sum of the ten-year and three-month yields ($2y(24) - y(3) - y(120)$). This factor can be either positive or negative.

2.2 Time-varying shape parameter

We extend the NS3F model by allowing the shape parameter λ_t to vary over time. The logarithm of λ_t will be added as a fourth time-varying factor to ensure positivity of λ_t. The $K = 4$ time-varying factors are captured in $\theta_t = (\beta_{1t}, \beta_{2t}, \beta_{3t}$. The observation equation of the four-factor Nelson-Siegel model (NS4F model) becomes

$$y_t = \Lambda(\lambda_t)\beta_t + \epsilon_t, \qquad \epsilon_t \sim \text{NID}(0, \Sigma_\epsilon). \tag{4}$$

We again assume that the dynamics of the factors are captured by a VAR[1] model given by

$$\theta_{t+1} = \omega + \Phi\theta_t + \eta_{t+1}, \qquad \eta_t \sim \text{NID}(0, \Sigma_\eta). \tag{5}$$

Compared to the VAR model in Equation 3, the dimensions of ω, Φ, η_t and Σ_η change as $K = 4$ in the NS4F model.

3 ProPar filter

This section explains how the proximal-parameter (ProPar) filtering method from Lange et al. (2022) works and applies it to the Nelson-Siegel framework from Section 2. First, we only consider the NS3F model in which the level, slope, and curvature are the three time-varying factors (Section 3.1). Thereafter, Section 3.2 proclaims the estimation procedure of the ProPar filtering method applied in the NS4F model, where the time-varying shape parameter causes nonlinearities in the filter.

3.1 ProPar filter in the NS3F model

Lange et al. (2022) propose the ProPar filtering method, which can approximate the true observation density $p_o(\cdot|\theta_t^o, \psi^o, \mathcal{F}_{t-1})$ of the variable of interest y_t, observed at different times, $t = 1, ..., T$. The variable θ_t^o is a time-varying parameter vector that can take values in the parameter space Θ^o, while ψ is a vector of static parameters, and $\mathcal{F}_{t-1}$ represents the information available at time $t-1$. The latter two symbols are suppressed in the density for readability. The postulated density $p(\cdot|\theta_t)$ contains the $K \times 1$ vector of time-varying parameters θ_t that is in the non-empty convex parameter space $\Theta \subseteq \mathbb{R}^K$. The predicted and updated parameters are denoted as $\theta_{t|t-1}$ and $\theta_{t|t}$ respectively, and stand for the estimates of θ_t based on the information available at times $t-1$ and t respectively.

The main challenge of any filtering method is determining how the predicted factors $\theta_{t|t-1}$ should be updated after observing y_t to get the updated factors $\theta_{t|t}$. In the ProPar filtering method, the factors are updated by maximizing the logarithmic observation density $\log p(y_t|\cdot)$ while subjecting it to a weighted l_2 penalty centered at the prediction $\theta_{t|t-1}$. This approach satisfies the two criteria of being in accordance with the likelihood (i.e., the parameter update yields an improved fit) and regularizing the deviation of the update from the prediction. The parameter update becomes

$$\theta_{t|t} := \underset{\theta\in\Theta}{\operatorname{argmax}}\, f(\theta|y_t, \theta_{t|t-1}, P_t), \tag{6}$$

where

$$f(\theta|y_t, \theta_{t|t-1}, P_t) := \log p(y_t|\theta) - \frac{1}{2}||\theta - \theta_{t|t-1}||^2_{P_t}. \tag{7}$$

$f(\theta|y_t, \theta_{t|t-1}, P_t)$ represents the regularized log-likelihood contribution in which $||x||^2_{P_t} = xP_tx$ is the squared l_2 norm with respect to a $K \times K$ positive-definite penalty matrix P_t. In the NS3F model, we have that $K = 3$. Note that the vector of time-varying factors θ_t only contains the level, slope, and curvature factors in the NS3F model, which are also represented by the vector β_t. However, for generality, we stick to the notation with θ_t as the same logic applies when more factors are added to the θ_t vector.

To determine the validity of the use of the ProPar filter for this NS3F model, we evaluate the concavity of the logarithmic observation density. The Hessian matrix of the logarithmic observation density is given in Appendix B.2. The negative definiteness of the Hessian implies concavity of the observation density in all β_t dimensions. Hence, the ProPar update step can be applied appropriately for the NS3F model.

The first-order condition of Equation 6 can be rewritten as

$$\begin{aligned} \theta_{t|t} &= \theta_{t|t-1} + H_t \nabla(y_t|\theta_{t|t}) \\ &= \theta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t|t}), \end{aligned} \tag{8}$$

where $H_t := P_t^{-1}$ is the learning-rate matrix and $\nabla(y_t|\theta_{t|t}) := (\partial \log p(y_t|\theta)/\partial\theta)|_{\theta=\theta_{t|t}}$ is the score vector which equals $\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t|t})$ as derived in the mathematical derivations in Appendix B.1. The evaluation of the score is at the updated $\theta_{t|t}$, not the prediction $\theta_{t|t-1}$ which is the case in explicit counterparts such as the dynamic conditional score (DCS) filtering method. As a result, Equation 8 acts as an implicit gradient method. The framework resembles implicit stochastic-gradient methods due to the random and stochastic nature of $\theta_{t|t}$, which is based on the observed data y_t.

As the observation equation in the NS3F model is linear in the three factors, the ProPar update step can be computed analytically. The ProPar update step becomes

$$\begin{aligned} \beta_{t|t} &= \beta_{t|t-1} + H_t \nabla(y_t|\theta_{t|t}) \\ &= \beta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t|t}) \\ &= \beta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1} y_t - H_t \Lambda' \Sigma_\epsilon^{-1}\Lambda\beta_{t|t} \\ \iff \qquad \big(I + H_t\Lambda'\Sigma_\epsilon^{-1}\Lambda\big)\beta_{t|t} &= \beta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1} y_t \\ \iff \qquad \beta_{t|t} &= \big(I + H_t\Lambda'\Sigma_\epsilon^{-1}\Lambda\big)^{-1}\big(\beta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1} y_t\big) \\ &= \big(I - H_t(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda H_t)^{-1}\Lambda'\Sigma_\epsilon^{-1}\Lambda\big)\big(\beta_{t|t-1} + H_t \Lambda' \Sigma_\epsilon^{-1} y_t\big) \qquad (9) \\ &= \beta_{t|t-1} + H_t(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda H_t)^{-1}\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t|t-1}) \\ &= \beta_{t|t-1} + H_t(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda H_t)^{-1}\nabla(y_t|\theta_{t|t-1}), \qquad (10) \end{aligned}$$

where I denotes the identity matrix. To get to Equation 9, we use the Woodbury matrix identity. In the derivation of Equation 10, we exchanged θ_t with β_t to enhance readability. Nevertheless, moving forward, we will revert to using the notation of θ_t. The result of Equation 10 is that the linear ProPar update is not an implicit gradient update anymore, which significantly reduces computation time.

We let penalty matrix P_t depend on predictive Fisher matrix. The predictive Fisher matrix in the NS3F model is defined as

$$\mathcal{I}_{t|t-1} = -\mathbb{E}\big[\mathcal{H}(y_t|\theta_t)\big|\theta_{t|t-1}\big] = \Lambda'\Sigma_\epsilon^{-1}\Lambda, \tag{11}$$

where $\mathcal{H}(y_t|\theta_t)$ denotes the Hessian matrix which is explicitly derived in Appendix B.2 and is given by $-\Lambda'\Sigma_\epsilon^{-1}\Lambda$. The considered penalty matrices in the three-factor model are given by

$$P_t = \rho_\beta \mathcal{I}_{t|t-1}^k = \rho_\beta[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^k, \qquad k \in \{0, 1\}, \tag{12}$$

where ρ_β denotes a penalty scalar. The two penalty matrices that we consider are essentially a scalar multiple of the identity matrix ($k = 0$) and a scalar multiple of the predictive Fisher matrix ($k = 1$). The predictive Fisher matrix has full rank as long as shape parameter λ is not too close to zero (resulting in the second column of Λ to converge to the first) or too large (resulting in the third column of Λ to converge to the second).

When we use a scalar multiple of the predictive Fisher matrix ($k = 1$) for penalty matrix P_t and the model is linear in the time-varying factors, which is the case in the NS3F model, we can rewrite the update step of the ProPar filter as follows:

$$\begin{aligned}
\theta_{t+1|t+1} &= \theta_{t+1|t} + H_t(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda H_t)^{-1}\nabla(y_t|\theta_{t|t-1}) \\
&= \theta_{t+1|t} + \rho_\beta^{-1}[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^{-1}(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda\rho_\beta^{-1}[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^{-1})^{-1}\nabla(y_t|\theta_{t|t-1}) \\
&= \theta_{t+1|t} + \rho_\beta^{-1}[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^{-1}(I + \rho_\beta^{-1}I)^{-1}\nabla(y_t|\theta_{t|t-1}) \\
&= \theta_{t+1|t} + (\rho_\beta + 1)^{-1}[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^{-1}\nabla(y_t|\theta_{t|t-1}) \\
&= \theta_{t+1|t} + \tilde{\rho}_\beta^{-1}[\Lambda'\Sigma_\epsilon^{-1}\Lambda]^{-1}\nabla(y_t|\theta_{t|t-1}) \\
&= \theta_{t+1|t} + \tilde{H}_t\nabla(y_t|\theta_{t|t-1}),
\end{aligned}$$

where $\tilde{H}_t$ is the learning-rate matrix with penalty scalar $\tilde{\rho}_\beta$. The last line of the derivation denotes the update step of the DCS filter (see Appendix C). Hence, when the model is linear in the time-varying factors, and we use a penalty matrix that is a scalar multiple of the predictive Fisher matrix, the ProPar filter and the DCS filter converge to the same filter, as rescaling can be done via the linear transformation in the penalty scalars $\rho_\beta^{ProPar} = 1 + \rho_\beta^{DCS}$.

To complement the update step, we need a prediction step that generates one-step-ahead forecasts. We assume the dynamics of the factors to be captured by the VAR[1] model in Equation 3.

Thus, we consider the prediction step to be the one-step-ahead forecast of that VAR[1] model:

$$\theta_{t+1|t} = \omega + \Phi\theta_{t|t}, \tag{13}$$

with $K \times 1$ vector of constants ω and $K \times K$ coefficient matrix Φ.

All remaining unknown static parameters are contained in $\psi = (\lambda, \mathrm{vec}(\Sigma_\epsilon)', \omega', \mathrm{vec}(\Phi)', \rho_\beta)'$ and are estimated by maximizing the log-likelihood expressed in terms of prediction errors

$$l(\psi) = -\frac{NT}{2}\log(2\pi) - \frac{T}{2}\sum_{i=1}^{N}\log\gamma_i - \frac{1}{2}\sum_{t=1}^{T}(y_t - \Lambda\beta_{t|t-1})'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t|t-1}), \tag{14}$$

where γ_i denotes the i^{th} eigenvalue of Σ_ϵ. Note that for the second term in Equation 19, we use the sum of the logarithms of eigenvalues of Σ_ϵ instead of the logarithms of the determinant of Σ_ϵ, as this term is more stable.

We compare the ProPar filter to three benchmark filtering methods, namely the dynamic conditional score (DCS) filter, the Kalman filter, and the GAS-Kalman filter, which is a self-devised version of the Kalman filter that is based on GAS-dynamics in the factors instead of a Gaussian-dynamics. The update and prediction steps of all filters are given in Table 1. As can be seen, the ProPar, DCS and Kalman filters have the same prediction steps for the factors. The update step differs, although the structures are similar. In the GAS-Kalman filter, the score is used in both the update and the prediction step. The GAS specification that we assume for the factor dynamics implies that the score evaluated at today's states contributes to next period's states, hence the presence of the score in the prediction step. Detailed derivations and explanations of the considered benchmark filtering methods can be found in Appendix C. Other benchmark models that we use for comparison are straightforward models such as a first-order vector autoregressive (VAR[1]) model on the factors and a 'naive forecasting' model which forecasts the one-step-ahead factors as today's fitted factors.

Table 1: Filtering methods and models used in the four-factor Nelson-Siegel framework

ProPar filter	$\beta_{t\vert t} = \beta_{t\vert t-1} + H_t(I + \Lambda'\Sigma_\epsilon^{-1}\Lambda H_t)^{-1}\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t\vert t-1})$ $\beta_{t+1\vert t} = \omega + \Phi\beta_{t\vert t}$
DCS filter	$\beta_{t\vert t} = \beta_{t\vert t-1} + H_t\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t\vert t-1})$ $\beta_{t+1\vert t} = \omega + \Phi\beta_{t\vert t}$
Kalman filter	$\beta_{t\vert t} = \beta_{t\vert t-1} + B_{t\vert t-1}\Lambda'(\Lambda B_{t\vert t-1}\Lambda' + \Sigma_\epsilon)^{-1}(y_t - \Lambda\beta_{t\vert t-1})$ $B_{t\vert t} = B_{t\vert t-1} - B_{t\vert t-1}\Lambda'(\Lambda B_{t\vert t-1}\Lambda' + \Sigma_\epsilon)^{-1}\Lambda B_{t\vert t-1}$ $\beta_{t+1\vert t} = \omega + \Phi\beta_{t\vert t}$ $B_{t+1\vert t} = \Phi B_{t\vert t}\Phi' + \Sigma_\eta$
GAS-Kalman filter	$\beta_{t\vert t} = \beta_{t\vert t-1} + B_{t\vert t-1}\Lambda'(\Lambda B_{t\vert t-1}\Lambda' + \Sigma_\epsilon)^{-1}(y_t - \Lambda\beta_{t\vert t-1})$ $B_{t\vert t} = B_{t\vert t-1} - B_{t\vert t-1}\Lambda'(\Lambda B_{t\vert t-1}\Lambda' + \Sigma_\epsilon)^{-1}\Lambda B_{t\vert t-1}$ $\beta_{t+1\vert t} = \omega + \Phi\beta_{t\vert t} + H_t\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_{t\vert t})$ $B_{t+1\vert t} = \Phi B_{t\vert t}\Phi' + H_t\Lambda'\Sigma_\epsilon^{-1}\Lambda B_{t\vert t}\Lambda'\Sigma_\epsilon^{-1}\Lambda H_t'$
VAR[1] model	$\beta_{t\vert t} = \text{argmin}_\beta(y_t - \Lambda\beta)'(y_t - \Lambda\beta)$ $\beta_{t+1\vert t} = \omega + \Phi\beta_{t\vert t}$
Naive forcasting model	$\beta_{t\vert t} = \text{argmin}_\beta(y_t - \Lambda\beta)'(y_t - \Lambda\beta)$ $\beta_{t+1\vert t} = \beta_{t\vert t}$

Note: The table reports the explicit equations of the ProPar filter, three benchmark filtering methods (the dynamic conditional score (*DCS*) filter, the Kalman filter and a version of the Kalman filter that is based on GAS specified factor dynamics (*GAS-Kalman*)), and two other benchmark models (a first-order vector autoregressive (*VAR[1]*) model and a naive forecasting model) in the three-factor Nelson-Siegel framework. The time-varying factors are the level, slope, and curvature factor that are contained in β_t. Detailed derivation of the ProPar filter can be found in Section 3, and derivations of the DCS, Kalman and GAS-Kalman filters can be found in Appendix C.

3.2 ProPar filter in the NS4F model

In this section, we apply the ProPar filtering method to the NS4F model. The four time-varying factors are modeled through $\theta_t = (\beta_t', \log\lambda_t)'$. The main challenge compared to the NS3F model, is that the observation equation becomes nonlinear in the factors. This nonlinearity means that the update step cannot be solved analytically. Therefore we have to perform a numerical optimization at each time step to compute the factor updates from Equation 6.

Again, we let penalty matrix P_t depend on the predictive Fisher matrix. The predictive Fisher matrix in the NS4F model is given by

$$\mathcal{I}_{t|t-1} = -\mathbb{E}\big[\mathcal{H}(y_t|\theta_t)\big|\theta_{t|t-1}\big] \tag{15}$$

$$= \begin{bmatrix} \Lambda(\lambda_{t|t-1})'\Sigma_\epsilon^{-1}\Lambda(\lambda_{t|t-1}) & \Lambda(\lambda_{t|t-1})\Sigma_\epsilon^{-1}\dot{\Lambda}(\lambda_{t|t-1})\beta_{t|t-1} \\ \beta_{t|t-1}'\dot{\Lambda}(\lambda_{t|t-1})'\Sigma_\epsilon^{-1}\Lambda(\lambda_{t|t-1}) & \beta_{t|t-1}'\dot{\Lambda}(\lambda_{t|t-1})'\Sigma_\epsilon^{-1}\dot{\Lambda}(\lambda_{t|t-1})\beta_{t|t-1} \end{bmatrix}, \tag{16}$$

where $\mathcal{H}(y_t|\theta_t)$ denotes the Hessian matrix which is explicitly derived in Appendix B.2, and $\dot{\Lambda}$ is the derivative of Λ with respect to $\log \lambda_t$. The considered penalty matrices in the NS4F model are given by

$$P_t = \begin{bmatrix} \rho_\beta I_3 & 0 \\ 0 & \rho_\lambda \end{bmatrix} \mathcal{I}_{t|t-1}^k, \qquad k \in \{0,1\}. \tag{17}$$

where ρ_λ represents a penalty scalar that only penalizes the $\log \lambda_t$ factor. Again, the two penalty matrices that we consider are essentially a matrix multiple of the identity matrix ($k = 0$) and a matrix multiple of the predictive Fisher matrix ($k = 1$).

For the prediction step, we consider the one-step-ahead prediction of the VAR[1] model in Equation 5 that is assumed to capture the dynamics of θ_t. This results in the following linear first-order specification:

$$\theta_{t+1|t} = \omega + \Phi\theta_{t|t}. \tag{18}$$

Compared to Equation 13, the dimensions of ω and Φ change as θ_t consist of four factors instead of three.

The remaining unknown static parameters are $\psi = (\text{vec}(\Sigma_\epsilon)', \omega', \text{vec}(\Phi)', \rho_\beta, \rho_\lambda)'$, which are estimated by maximizing the log-likelihood in terms of prediction errors. This log-likelihood is given by

$$l(\psi) = -\frac{NT}{2}\log(2\pi) - \frac{T}{2}\sum_{i=1}^{N}\log\gamma_i - \frac{1}{2}\sum_{t=1}^{T}\big(y_t - \Lambda(\lambda_{t|t-1})\beta_{t|t-1}\big)'\Sigma_\epsilon^{-1}\big(y_t - \Lambda(\lambda_{t|t-1})\beta_{t|t-1}\big). \tag{19}$$

We compare the ProPar filter to three benchmark filtering methods, namely the dynamic conditional score (DCS) filter, the Kalman filter, and the self-devised GAS-Kalman filter. An overview of the update and prediction steps can be found in Table 2. Appendix C provides detailed derivations of the update and prediction steps for these benchmark filters. Table 2 also contains the other benchmark models that are considered (VAR[1] model and 'Naive Forecasting' model).

Table 2: Filtering methods and models used in the four-factor Nelson-Siegel framework

ProPar filter	$\theta_{t\vert t} = \theta_{t\vert t-1} + H_t \nabla(y_t \vert \theta_{t\vert t})$ $\theta_{t+1\vert t} = \omega + \Phi\theta_{t\vert t}$
DCS filter	$\theta_{t\vert t} = \theta_{t\vert t-1} + H_t \nabla(y_t \vert \theta_{t\vert t-1})$ $\theta_{t+1\vert t} = \omega + \Phi\theta_{t\vert t}$
Kalman filter	$\theta_{t\vert t} = \theta_{t\vert t-1} + Q_{t\vert t-1}\breve{Z}_t'(\breve{Z}_t Q_{t\vert t-1}\breve{Z}_t' + \Sigma_\epsilon)^{-1}\big(y_t - Z(\theta_{t\vert t-1})\big)$ $Q_{t\vert t} = Q_{t\vert t-1} - Q_{t\vert t-1}\breve{Z}_t'(\breve{Z}_t Q_{t\vert t-1}\breve{Z}_t' + \Sigma_\epsilon)^{-1}\breve{Z}_t Q_{t\vert t-1}$ $\theta_{t+1\vert t} = \omega + \Phi\theta_{t\vert t}$ $Q_{t+1\vert t} = \Phi Q_{t\vert t}\Phi' + \Sigma_\eta$
GAS-Kalman filter	$\theta_{t\vert t} = \theta_{t\vert t-1} + Q_{t\vert t-1}\breve{Z}_t'(\breve{Z}_t Q_{t\vert t-1}\breve{Z}_t' + \Sigma_\epsilon)^{-1}\big(y_t - Z(\theta_{t\vert t-1})\big)$ $Q_{t\vert t} = Q_{t\vert t-1} - Q_{t\vert t-1}\breve{Z}_t'(\breve{Z}_t Q_{t\vert t-1}\breve{Z}_t' + \Sigma_\epsilon)^{-1}\breve{Z}_t Q_{t\vert t-1}$ $\theta_{t+1\vert t} = \omega + \Phi\theta_{t\vert t} + H_t \nabla(y_t \vert \theta_{t\vert t})$ $Q_{t+1\vert t} = \Phi Q_{t\vert t}\Phi' + H_t \nabla^2(y_t \vert \theta_{t\vert t}) Q_{t\vert t} \nabla^2(y_t \vert \theta_{t\vert t})' H_t'$
VAR[1] model	$\theta_{t\vert t} = \text{argmin}_\theta \big(y_t - \Lambda(\lambda)\beta\big)'\big(y_t - \Lambda(\lambda)\beta\big)$ $\theta_{t+1\vert t} = \omega + \Phi\theta_{t\vert t}$
Naive forcasting model	$\theta_{t\vert t} = \text{argmin}_\theta \big(y_t - \Lambda(\lambda)\beta\big)'\big(y_t - \Lambda(\lambda)\beta\big)$ $\theta_{t+1\vert t} = \theta_{t\vert t}$

Note: The table reports the explicit equations of the ProPar filter, three benchmark filtering methods (the dynamic conditional score (*DCS*) filter, the Kalman filter and a version of the Kalman filter that is based on GAS specified factor dynamics (*GAS-Kalman*)), and two other benchmark models (a first-order vector autoregressive (*VAR[1]*) model and a naive forecasting model) in the four-factor Nelson-Siegel framework. The time-varying factors are the level, slope, and curvature factor and shape parameter λ_t, which are all contained in θ_t. Detailed derivation of the ProPar filter can be found in Section 3, and derivations of the DCS, Kalman and GAS-Kalman filters can be found in Appendix C.

4 Simulation experiment

To evaluate whether the ProPar filtering method and the benchmark methods can estimate and forecast yield curves in the Nelson-Siegel framework, we first perform a simulation study. In the simulation study, we generate yield data following two different model specifications. We start with an observation-driven model, in which the yield data is generated following a generalized autoregressive score (GAS) specification (Section 4.1). It would be unfair to limit the comparison to only this specification, as the Kalman filter exhibits a greater degree of misspecification in this setup than filters that are based on score-driven factor dynamics. Therefore, we also compare estimation and forecasting performances in a state-space Nelson-Siegel framework, in which the time-varying factors follow a Gaussian autoregressive model (Section 4.2). For both model specifications, the ProPar filter is misspecified and is expected to be beaten by the correctly specified benchmark filtering method.

4.1 Generalized autoregressive score specification

In the GAS specification, we generate yield data according

$$\begin{aligned} y_t &= \Lambda(\lambda_t)\beta_t + \epsilon_t, & \epsilon_t &\sim \text{NID}(0, \Sigma_\epsilon), \\ \theta_{t+1} &= \omega + \Phi\theta_t + H_t \nabla(y_t|\theta_t), \end{aligned}$$

in which θ_t consists of the time-varying factors, which is β_t for the three-factor (NS3F) model and additionally $\log \lambda_t$ in the four-factor (NS4F) model. $\nabla(y_t|\theta_t)$ denotes the score vector of θ at time t, which is derived in Appendix B.1. We choose $\theta_0 = \mathbf{0}$ as a starting point and H_t to be the inverse of the penalty matrix from Equation 12 (NS3F model) and Equation 17 (NS4F model) with $k = 0$. The true parameter values can be found in the first columns of Table 3 and 4 for the NS3F and NS4F models respectively. In the NS3F model, λ is set to 0.05, which is close to the value of 0.0609 that is proposed by Diebold and Li (2006). For convenience, the covariance matrix Σ_ϵ is set to 0.2 times the identity matrix. This setting is not very realistic when we compare it to real yield data, as yield data is usually highly correlated and hence has positive covariances. The parameters ω and Φ are chosen such that the long term mean of the factors, which is given by $\mu = (I_K - \Phi)^{-1}\omega$, equals $(5, -5, -1)'$ in the three-factor case, and $(5, -5, -1, -3)'$ in the four-factor case. The level factor is generally positive, the slope factor is usually negative and the curvature factor can both be positive or negative (Diebold and Li, 2006). The logarithm of the shape parameter is in the long-term centered around -3 which is also a realistic value. Therefore, we consider this to be appropriate values for the long-term means. The penalty scalars ρ_β and ρ_λ are chosen arbitrarily and quite high, as low values result in large outliers using this setup.

In the simulation setup, we choose the set of maturities $\tau = \{6, 12, 24, 60\}$ of size $N = 4$. Moreover, we set $T = 1000$ where the estimation is done on a train set consisting of the first 800 time points, and the out-of-sample forecasting performance is evaluated on the final 20% of the data. We perform 250 replications, after which the parameter estimates, log-likelihood, and mean squared forecast error (MSFE) are computed as the average over the 250 iterations × 200 observations. We refer to the correctly specified DCS filter where the true parameter values are used as the 'true model'. We have to note that it is computationally intensive to estimate 21 (for the NS3F model) and 26 (for the NS4F model) parameters. This also explains the choice for the relatively low sample size and number of replications. Especially in the four-factor framework, in which the the proximal parameter update needs a numerical optimization at each time step, the estimation problem becomes computationally heavy. Because of limitations in time and computational power, we cannot ensure that the static parameter estimates correspond to the global optimum of the log-likelihood. This can affect the parameter estimation accuracy and thus the state filtering performance.

The estimation results and forecasting performance of the ProPar filter and the benchmark filters in the NS3F and the NS4F model are displayed in Tables 3 and 4 respectively. The tables suggest that the filters based on score-driven factor dynamics are better in estimating the parameters compared to the Kalman filter as they generally provide both smaller biases and standard deviations. This makes sense as the data-generating process also implies that the factor dynamics are score-driven. The biggest differences in parameter estimation among the score-driven filters are the estimates for the penalty parameters ρ_β (and ρ_λ in the NS4F model). It is coherent that the DCS filter estimates this penalty term close to its true value, as the data is correctly specified for this filter. The ProPar filter and the GAS-Kalman filter can, to some extent, correct for misspecification via the penalty scalars.

Next to that, the DCS filter and the ProPar filter provide the best model fit for both the NS3F and the NS4F model. They both beat the correctly specified DCS filter with true parameter values in terms of model fit. This means the true residuals are bigger than the residuals that result from the fitted values of these filters. Nevertheless, the differences with the other filters are small. In the NS3F model, the DCS filter (best model fit) beats the Kalman filter (worst model fit) by only 1.22% and in the NS4F model, the DCS filter (best model fit) beats the Kalman filter (worst model fit) by only 2.44%.

Table 3: Simulation results of various filtering methods in the three-factor Nelson-Siegel model using generalized autoregressive score generated data.

	Truth	ProPar filter		DCS filter		Kalman filter		Kalman filter (OD)	
λ	0.05	0.0338	(0.0164)	0.0473	(0.0203)	0.0197	(0.0107)	0.0444	(0.0225)
Σ_{11}	0.2	0.1988	(0.0088)	0.1989	(0.0089)	0.1644	(0.0092)	0.1892	(0.0088)
Σ_{22}	0.2	0.1998	(0.0102)	0.1999	(0.0102)	0.1677	(0.0105)	0.1910	(0.0102)
Σ_{33}	0.2	0.1992	(0.0097)	0.1992	(0.0097)	0.1703	(0.0098)	0.1914	(0.0096)
Σ_{44}	0.2	0.2005	(0.0094)	0.1997	(0.0093)	0.1732	(0.0094)	0.1933	(0.0093)
Σ_{12}	0	-0.0013	(0.0073)	-0.0011	(0.0073)	-0.0346	(0.0076)	-0.0104	(0.0072)
Σ_{13}	0	-0.0016	(0.0069)	-0.0012	(0.0070)	-0.0326	(0.0069)	-0.0099	(0.0067)
Σ_{14}	0	-0.0007	(0.0067)	0.0000	(0.0067)	-0.0265	(0.0072)	-0.0075	(0.0066)
Σ_{23}	0	-0.0009	(0.0071)	-0.0007	(0.0071)	-0.0311	(0.0077)	-0.0090	(0.0071)
Σ_{24}	0	-0.0006	(0.0067)	-0.0002	(0.0067)	-0.0268	(0.0069)	-0.0074	(0.0066)
Σ_{34}	0	0.0002	(0.0070)	0.0002	(0.0070)	-0.0267	(0.0076)	-0.0068	(0.0069)
ω_1	0.1	0.1579	(0.4509)	0.1729	(0.3549)	0.3236	(1.7324)	0.2167	(0.3680)
ω_2	-0.1	-0.1578	(0.4186)	-0.1696	(0.3267)	-0.1108	(1.5215)	-0.1943	(0.3462)
ω_3	-0.02	-0.0276	(0.8009)	-0.1330	(0.6343)	2.6113	(3.0511)	-0.1658	(0.7481)
Φ_{11}	0.98	0.9302	(0.1330)	0.9585	(0.0566)	0.8966	(0.1329)	0.9497	(0.0563)
Φ_{22}	0.98	0.9362	(0.0815)	0.9467	(0.0691)	0.9201	(0.0844)	0.9402	(0.0739)
Φ_{33}	0.98	0.7699	(0.3169)	0.8051	(0.2919)	0.6062	(0.3242)	0.7932	(0.2785)
Φ_{12}	0	0.0269	(0.0881)	0.0028	(0.0493)	0.0430	(0.0858)	0.0197	(0.0532)
Φ_{13}	0	0.0228	(0.1343)	0.0099	(0.1065)	-0.0147	(0.1802)	0.0376	(0.1135)
Φ_{23}	0	-0.0171	(0.1504)	0.0014	(0.1343)	0.0202	(0.1782)	-0.0060	(0.1454)
ρ_β	50	22.2988	(3.9727)	50.2614	(5.5820)			73.6125	(7.8607)
σ_β						0.1038	(0.0089)		
$l(\hat{\psi})$	-1967.7	-1957.5		-1956.2		-1980.0		-1977.8	
MSFE	0.2009	0.2041		0.2043		0.2019		0.2020	

Note: The table reports the estimation results and forecasting performance of the ProPar filter, the DCS filter, the Kalman filter and the observation-driven variant of the Kalman filter (OD) in the three-factor Nelson-Siegel framework. The data is generated using a generalized autoregressive score model. For each static parameter in the model, the true value, the estimates and its standard deviation (between brackets) are given. Also, the log-likelihood value on the train-sample, $l(\hat{\psi})$, and the mean squared forecast error (MSFE) on the test data are displayed for each filtering method. The 'true' log-likelihood and MSFE are those of the DCS filter where the 'true' parameter values are used. The estimates, log-likelihood values and MSFEs are obtained by taking the average over 250 iterations, and the standard deviation is the standard deviation over 250 iterations. The 250 simulated data sets each consist of $T = 1000$ observations and a train/test-split of 80/20 is used.

Table 4: Simulation results of various filtering methods in the four-factor Nelson-Siegel model using generalized autoregressive score generated data.

	Truth	ProPar filter		DCS filter		Kalman filter		Kalman filter (OD)	
Σ_{11}	0.2	0.2060	(0.0089)	0.1991	(0.0084)	0.1517	(0.0096)	0.1838	(0.0105)
Σ_{22}	0.2	0.2031	(0.0093)	0.2001	(0.0092)	0.1517	(0.0099)	0.1808	(0.0121)
Σ_{33}	0.2	0.2017	(0.0085)	0.1992	(0.0089)	0.1450	(0.0103)	0.1763	(0.0115)
Σ_{44}	0.2	0.2039	(0.0093)	0.1992	(0.0086)	0.1455	(0.0101)	0.1774	(0.0124)
Σ_{12}	0	0.0020	(0.0069)	-0.0011	(0.0070)	-0.0471	(0.0074)	-0.0183	(0.0083)
Σ_{13}	0	-0.0024	(0.0065)	-0.0010	(0.0066)	-0.0440	(0.0066)	-0.0192	(0.0086)
Σ_{14}	0	-0.0039	(0.0066)	0.0001	(0.0063)	-0.0353	(0.0067)	-0.0168	(0.0077)
Σ_{23}	0	-0.0017	(0.0064)	-0.0008	(0.0067)	-0.0499	(0.0071)	-0.0220	(0.0085)
Σ_{24}	0	-0.0033	(0.0063)	-0.0005	(0.0063)	-0.0432	(0.0065)	-0.0201	(0.0077)
Σ_{34}	0	-0.0004	(0.0059)	-0.0005	(0.0066)	-0.0510	(0.0072)	-0.0227	(0.0089)
ω_1	0.1	0.1151	(0.0348)	0.1290	(0.0876)	0.4785	(0.5835)	0.1426	(0.2309)
ω_2	-0.1	-0.0856	(0.0395)	-0.0954	(0.1557)	-0.0146	(0.6390)	-0.1945	(0.3120)
ω_3	-0.02	0.0556	(0.0787)	0.0181	(0.2523)	0.3345	(0.7721)	-0.0511	(0.5715)
ω_4	-0.06	-0.1142	(0.0619)	-0.1346	(0.2748)	-0.6536	(0.9657)	-0.1121	(0.2284)
Φ_{11}	0.98	0.9737	(0.0083)	0.9575	(0.0579)	0.8844	(0.1081)	0.9693	(0.0234)
Φ_{22}	0.98	0.9672	(0.0248)	0.9418	(0.0878)	0.7667	(0.1490)	0.9430	(0.0608)
Φ_{33}	0.98	0.9727	(0.0312)	0.9068	(0.1775)	0.7602	(0.2422)	0.8812	(0.1861)
Φ_{44}	0.98	0.9625	(0.0290)	0.9245	(0.1312)	0.6667	(0.2517)	0.9579	(0.0395)
Φ_{12}	0	-0.0053	(0.0107)	-0.0162	(0.0412)	-0.0929	(0.0843)	-0.0009	(0.0272)
Φ_{13}	0	0.0046	(0.0088)	-0.0108	(0.0729)	-0.0025	(0.0851)	-0.0003	(0.0490)
Φ_{14}	0	0.0101	(0.0125)	0.0116	(0.0482)	0.1358	(0.1406)	0.0022	(0.0228)
Φ_{23}	0	0.0032	(0.0071)	-0.0009	(0.1031)	0.0175	(0.1877)	0.0276	(0.0985)
Φ_{24}	0	0.0117	(0.0242)	0.0286	(0.0744)	0.2055	(0.1554)	0.0043	(0.0351)
Φ_{34}	0	-0.0043	(0.0095)	0.0135	(0.1090)	0.0557	(0.1538)	-0.0003	(0.0498)
ρ_β	50	11.3805	(5.2538)	49.5636	(6.1272)			87.9376	(13.8258)
ρ_λ	50	17.2566	(7.2915)	49.8708	(8.1223)			89.0554	(16.1386)
σ_β						0.1488	(0.0150)		
σ_λ						0.0956	(0.0383)		
$l(\hat{\psi})$	-1967.8	-1989.6		-1955.0		-1998.9		-1998.3	
MSFE	0.2009	0.2056		0.2040		0.2040		0.2039	

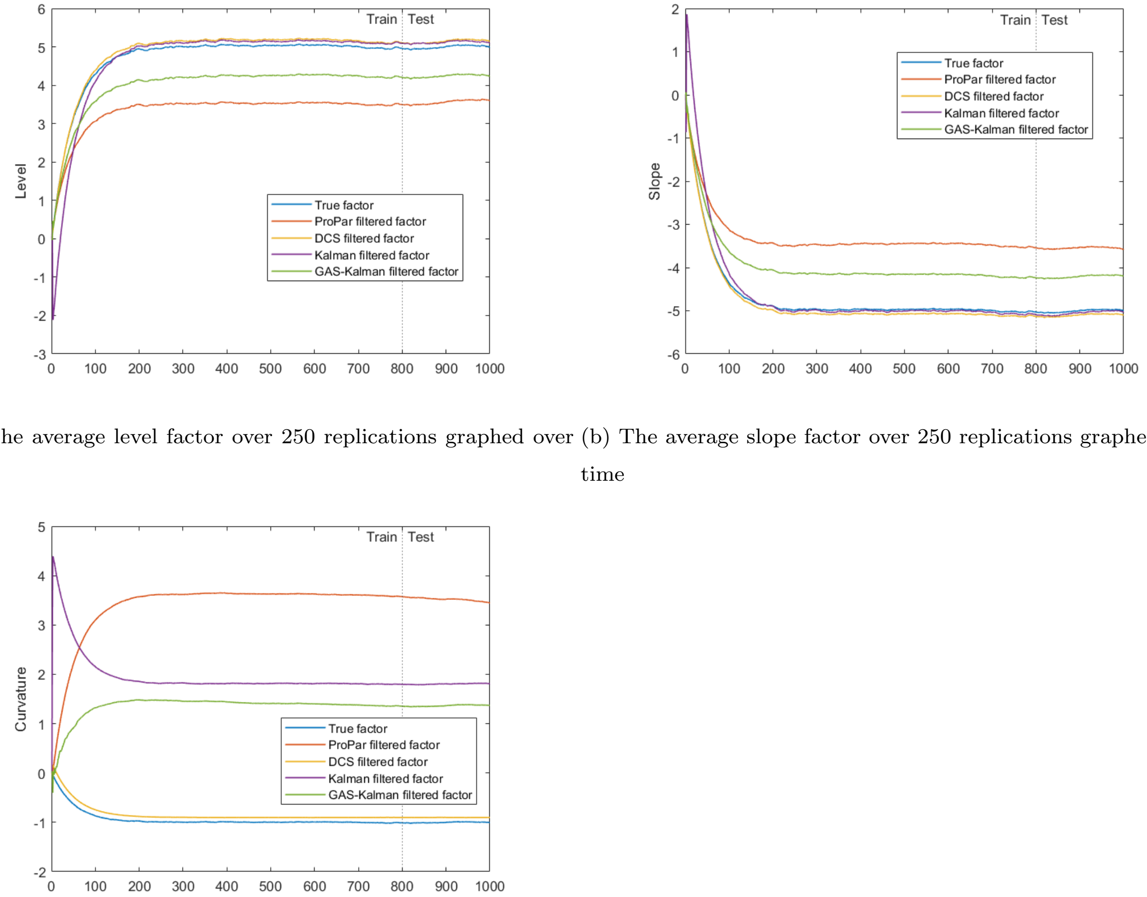
Note: The table reports the estimation results and forecasting performance of the ProPar filter, the DCS filter, the Kalman filter and the observation-driven variant of the Kalman filter (OD) in the four-factor Nelson-Siegel framework. The data is generated using a generalized autoregressive score model. For each static parameter in the model, the true value, the estimates and its standard deviation (between brackets) are given. Also, the log-likelihood value on the train-sample, $l(\hat{\psi})$, and the mean squared forecast error (MSFE) on the test data are displayed for each filtering method. The 'true' log-likelihood and MSFE are those of the DCS filter where the 'true' parameter values are used. The estimates, log-likelihood values and MSFEs are obtained by taking the average over 250 iterations, and the standard deviation is the standard deviation over 250 iterations. The 250 simulated data sets each consist of $T = 1000$ observations and a train/test-split of 80/20 is used.

Furthermore, both versions of the Kalman filter provide accurate yield forecasts in both the NS3F and the NS4F models, as their MSFEs are not far from the MSFE of the correctly specified filter. The ProPar filter and the DCS filter generate the least accurate yield forecasts in the NS3F model, and in the NS4F model the ProPar filter has the highest MSFE. However, its predictive performance is not too far from the Kalman filter's, as the MSFE of the ProPar filter is only 1.09% and 0.83% higher than that of the Kalman filter. The mean squared error of the yield forecasts of the VAR[1] model and the naive forecast model are displayed in Table 17 in Appendix D. We see that the forecasts of these rather simple benchmark models are less accurate than those of the more advanced filters form Tables 3 and 4.

Next to the estimation and yield forecasting performances, we compare the filtering methods based on filtering performance. Figures 1 and 2 graph the averaged, filtered factor values for all filtering methods over time of the NS3F and NS4F models respectively. In both models, we can see that, on average, the DCS filter is able to accurately estimate the level, slope, and curvature factors, as well as the shape parameter in the NS4F model. This is not surprising as the DCS filter is correctly specified in this setup. The Kalman filter estimates the level and slope factors in the NS3F model accurately, but is not accurate in estimating the curvature factor or any of the factors in the NS4F model. The ProPar method is the least effective among the filtering methods for factor filtration. This can partly be explained by its estimate for shape parameter λ in the NS3F model and its deviation in the $\log \lambda_t$ sequence in the NS4F model. A tiny error in the λ estimate results in different factor loadings, and hence different estimates for the level, slope, and curvature factors. Deviations in the shape parameter can cancel out deviations in the other factors to ensure an optimal yield fit. The GAS-Kalman filter is not filtering accurately in the NS3F model, but generates accurate factor estimates in the NS4F model.

These findings are confirmed by the mean and standard deviation of the mean squared forecast errors of the factors over the 250 replications, which are given in Tables 5 (NS3F model) and 6 (NS4F model). From these tables, we can also see that the ProPar filter is the most precarious among the replications in the NS3F model, while the Kalman filter performs most unstable in the NS4F model in terms of filtering the factors. Whereas the VAR[1] generates relatively good estimates of the factors (beating the ProPar filter in the NS3F model and the Kalman filter in the NS4F model), the naive forecast model is easy to beat.

In short, we find that for this score-driven Nelson-Siegel specification, the ProPar filter can fit the yield training data well, but provides less accurate out-of-sample yield forecasts compared to the Kalman filter. This finding suggests that the ProPar filter is relatively better able to capture the noises and patterns in the training data, whereas the Kalman filter better grasps the underlying true patterns that generalize well to unseen data. Also, the ProPar filter is not able to filter the effect of the factors accurately, and is significantly outperformed by the other benchmark filtering methods. The DCS filter does a remarkably good job in filtering the effect of the factors.

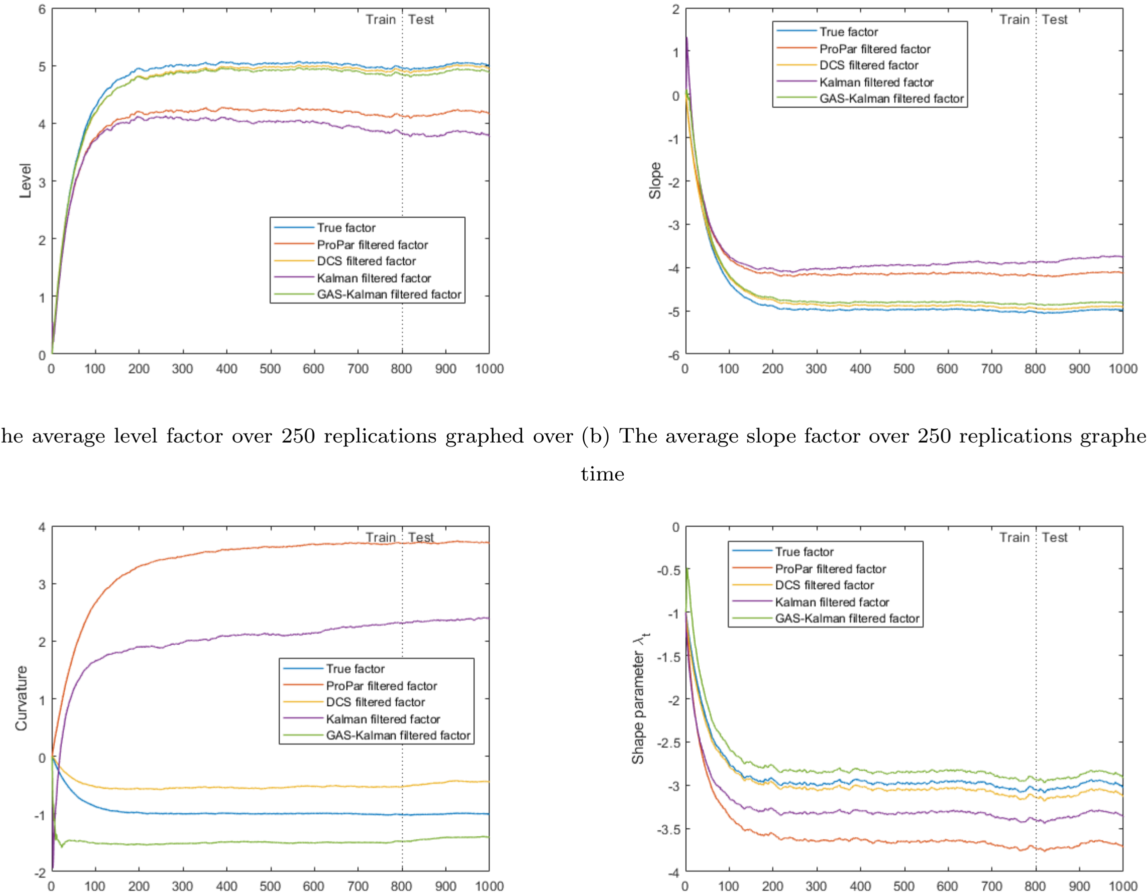


(a) The average level factor over 250 replications graphed over time

(b) The average slope factor over 250 replications graphed over time

(c) The average curvature factor over 250 replications graphed over time

Figure 1: **The replication-averaged level, slope, and curvature factors over time in the NS3F model with GAS-dynamics.** This figure shows the filtered level (panel a), slope (panel b), and curvature (panel c) factors in the NS3F model with GAS-dynamics averaged over the 250 replications. Next to the true simulated factor values, the figure shows the filtered factor values of the ProPar filter, the DCS filter, the Kalman filter and the GAS-Kalman filter.



(a) The average level factor over 250 replications graphed over time

(b) The average slope factor over 250 replications graphed over time

(c) The average curvature factor over 250 replications graphed over time

(d) The average shape parameter over 250 replications graphed over time

Figure 2: **The replication-averaged level, slope, and curvature factors over time in the NS4F model with GAS-dynamics.** This figure shows the filtered level (panel a), slope (panel b), and curvature (panel c) factors, and the logarithm of the shape parameter (panel d) in the NS4F model with GAS-dynamics averaged over the 250 replications. Next to the true simulated factor values, the figure shows the filtered factor values of the ProPar filter, the DCS filter, the Kalman filter and the GAS-Kalman filter.

Table 5: Filtering performance in the NS3F model with GAS-dynamics in terms of MSFE statistics.

	ProPar filter		DCS filter		Kalman filter	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	18.996	76.027	0.024	0.008	0.033	0.021
Slope factor	18.832	76.146	0.011	0.004	0.021	0.012
Curvature factor	76.351	220.491	0.024	0.018	7.883	0.346

	GAS-Kalman filter		VAR(1)		Naive forecast	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	5.159	22.864	0.228	0.073	3.402	0.342
Slope factor	5.199	22.928	0.387	0.137	1.827	0.175
Curvature factor	30.133	94.089	0.936	0.236	29.665	2.997

Note: The table reports the filtering performance, in terms of average mean squared forecast error over 250 replications (*Mean MSFE*) and standard deviation of the mean squared forecast errors from the 250 replications (*Stdev*) of the time-varying factors in the NS3F model. The NS3F model specification includes a GAS structure for the dynamics of the factors. The filters that are considered are the ProPar filter, the dynamic conditional score (*DCS*) filter, the Kalman filter, a version of the Kalman filter that assumes that the dynamics of the factors are GAS-specified (*GAS-Kalman*), the first-order vector autoregressive (*VAR[1]*) model and the naive forecast model. The mean squared errors at each repetition are calculated over the test sample which consists of the last 20% of the simulated data.

Table 6: Filtering performance in the NS4F model with GAS-dynamics in terms of MSFE statistics.

	ProPar filter		DCS filter		Kalman filter	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	2.644	9.449	0.269	1.601	21.720	136.231
Slope factor	2.732	9.611	0.292	1.655	21.969	137.156
Curvature factor	31.739	52.661	4.936	18.312	70.140	344.420
Shape parameter $\log \lambda_t$	0.573	0.449	0.142	0.411	0.349	0.661

	GAS-Kalman filter		VAR(1)		Naive forecast	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	0.126	0.743	8.109	3.737	160.890	107.438
Slope factor	0.169	0.822	17.371	5.862	261.250	120.629
Curvature factor	2.843	9.152	10.564	7.806	511.532	195.935
Shape parameter $\log \lambda_t$	0.117	0.174	0.388	0.265	0.978	0.311

Note: The table reports the filtering performance, in terms of average mean squared forecast error over 250 replications (*Mean MSFE*) and standard deviation of the mean squared forecast errors from the 250 replications (*Stdev*) of the time-varying factors in the NS4F model. The NS4F model specification includes a GAS structure for the dynamics of the factors. The filters that are considered are the ProPar filter, the dynamic conditional score (*DCS*) filter, the Kalman filter, a version of the Kalman filter that assumes that the dynamics of the factors are GAS-specified (*GAS-Kalman*), the first-order vector autoregressive (*VAR[1]*) model and the naive forecast model. The mean squared errors at each repetition are calculated over the test sample which consists of the last 20% of the simulated data.

4.2 State-space specification

To make a fair comparison between the Kalman filter and the filters that are based on score-driven factor dynamics, we evaluate the same filters and models on data generated by the state-space specification. The model is given by

$$\begin{aligned} y_t &= \Lambda(\lambda_t)\beta_t + \epsilon_t, & \epsilon_t &\sim \text{NID}(0, \Sigma_\epsilon), \\ \theta_{t+1} &= \omega + \Phi\theta_t + \eta_t, & \eta_t &\sim \text{NID}(0, \Sigma_\eta), \end{aligned}$$

in which η_t is a $K \times 1$ residual vector with zero mean and variance Σ_η. We choose Σ_η to be a diagonal matrix, with σ_β^2 on the diagonal entries that correspond to the level, slope, and curvature factors, and σ_λ^2 on the diagonal entry that corresponds to the fourth factor in the NS4F model. The true parameters have similar values compared to the simulation with GAS specification. The true values can be found in the first columns of Tables 7 (NS3F model) and 8 (NS4F model).

Along the same lines, we simulate 250 data sets using the specification above with a set of maturities $\tau = \{6, 12, 24, 60\}$ of size $N = 4$ and a sample size of $T = 1000$ with a train/test-split of 80/20. The parameter estimates, log-likelihood, and mean squared forecast error (MSFE) are determined as the average over the 250 replications. We refer to the correctly specified Kalman filter where the true parameter values are used as the 'true model'. We again stress that the number of static parameters is high (21 in the NS3F model and 26 in the NS4F model) making the parameter estimation computationally intensive. Due to time and computational power limits, we cannot ensure that the static parameter estimates correspond to the global optimum of the log-likelihood. This can affect the parameter estimation accuracy and thus the state filtering performance.

Tables 7 and 8 show the simulation results of respectively the NS3F model and the NS4F model using the state-space data-generating process. As expected, the Kalman filter has the highest accuracy in parameter estimation for both the NS3F and NS4F models, especially for the parameters in Σ_ϵ. Nonetheless, the parameter estimates for ω and Φ are equally, if not less, accurate in comparison to the estimates obtained through the other filtering methods.

Moreover, the Kalman filter generally has the best model fits, beating the true model in both the NS3F and the NS4F models. In the NS3F model, the ProPar filter even provides a better fit than the Kalman filter, but it has the worst model fit among all filters in the NS4F model. The DCS filter fits the data more or less equally well as the true model.

Next to fitting the train data best, the Kalman filter also provides the best yield forecasts in both models. In the NS3F model, its MSFE is smaller than the ProPar filter's MSFE by a tiny margin. In the NS4F model, the Kalman filter significantly outperforms all other filters in terms of forecasting. Based on the mean squared forecast error statistics for the other relatively basic benchmark models presented in Table 17 in Appendix D, it can be inferred that the advanced filtering methods in Tables 3 and 4 outperform them with respect to out-of-sample yield forecasting.

When we look at the averaged, filtered factor values, which are graphed in Figures 3 and 4 for the NS3F and NS4F models respectively, we see that the ProPar filter and the Kalman filter are best able to accurately estimate the factors. The ProPar filter provides the most accurate factor estimates in the NS3F model, but can be quite inaccurate in the NS4F model as it makes unusual jumps at some points in time. The inaccuracy of the ProPar filter in the NS4F model comes from the low shape parameter estimates. The long-term mean of the logarithm of the shape parameter seems to be around -3.5 (Figure 4), resulting in a different factor loading matrix Λ which affects the estimates of the other factors. The Kalman filter is most accurate in the NS4F model in terms of filtering, and does especially a good job when estimating the shape parameter λ_t. The DCS and GAS-Kalman filters are unable to precisely estimate the curvature factor in the NS3F model. This can be explained by the deviation in the estimate for shape parameter λ (see Table 7). As a result, the factors loadings are different, implying a different long-term mean for the curvature factor. Also, these filters exhibit a high degree of misspecification in this simulation setup.

Tables 9 and 10 present the mean and standard deviation of the mean squared forecast errors of the factors over the 250 replications for the NS3F and NS4F models respectively. These tables suggest that, on average, the factor estimation accuracy in the NS3F model does not differ substantially, except for the naive forecast model which is significantly less accurate. In the NS4F model, the relatively and unstable forecast error of the ProPar filter is noteworthy. The ProPar estimates are significantly less accurate than those of a simple VAR[1] model. The Kalman filter is clearly able to most accurately filter out the factors when a state-space specification is used.

Overall, for a state-space Nelson-Siegel specification, the Kalman filter is superior to the other filters in terms of model fit and yield forecasting accuracy and filtering performance. In the NS3F model, the differences with the other filtering methods are limited, but in the NS4F model the Kalman filter excels compared to the ProPar filter, the DCS filter, and the GAS-Kalman filter. This result is not surprising as the Kalman filter is the only filter that is correctly specified in this setup.

Table 7: Simulation results of various filtering methods in the three-factor Nelson-Siegel model using state-space generated data.

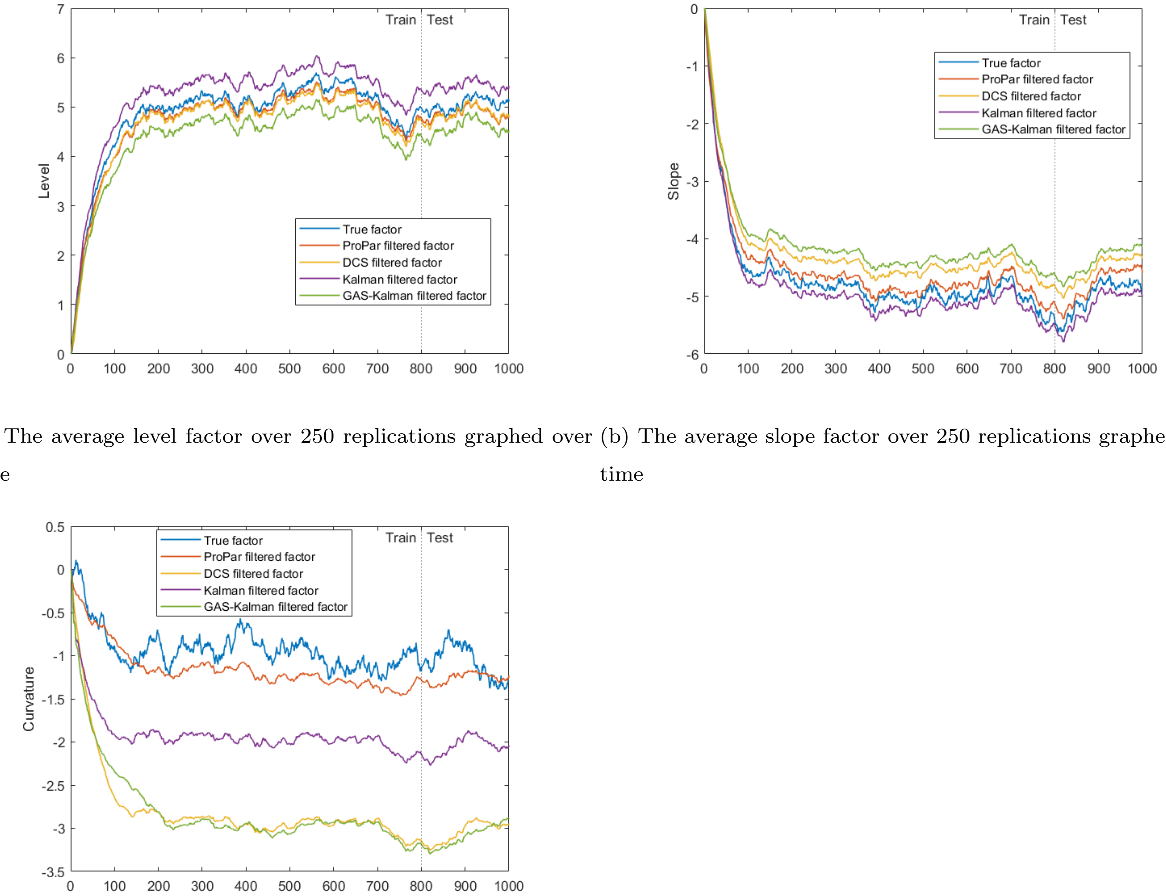
	Truth	ProPar filter		DCS filter		Kalman filter		Kalman filter (OD)	
λ	0.05	0.0576	(0.0150)	0.0835	(0.0152)	0.0541	(0.0151)	0.0896	(0.0099)
Σ_{11}	0.2	0.4143	(0.0199)	0.4459	(0.0228)	0.1981	(0.0168)	0.3660	(0.0218)
Σ_{22}	0.2	0.3860	(0.0172)	0.4185	(0.0221)	0.1986	(0.0155)	0.3456	(0.0212)
Σ_{33}	0.2	0.3652	(0.0161)	0.3917	(0.0190)	0.1988	(0.0140)	0.3283	(0.0188)
Σ_{44}	0.2	0.3573	(0.0159)	0.3697	(0.0165)	0.1992	(0.0148)	0.3209	(0.0174)
Σ_{12}	0	0.1959	(0.0152)	0.2285	(0.0197)	-0.0018	(0.0130)	0.1521	(0.0198)
Σ_{13}	0	0.1706	(0.0137)	0.2047	(0.0182)	-0.0009	(0.0113)	0.1338	(0.0184)
Σ_{14}	0	0.1337	(0.0132)	0.1708	(0.0175)	-0.0020	(0.0115)	0.1076	(0.0186)
Σ_{23}	0	0.1690	(0.0133)	0.1998	(0.0177)	-0.0018	(0.0112)	0.1317	(0.0173)
Σ_{24}	0	0.1399	(0.0127)	0.1700	(0.0163)	-0.0022	(0.0112)	0.1090	(0.0162)
Σ_{34}	0	0.1484	(0.0126)	0.1693	(0.0152)	-0.0016	(0.0106)	0.1124	(0.0147)
ω_1	0.1	0.1548	(0.2992)	0.1194	(0.0937)	0.1575	(0.1800)	0.1228	(0.0998)
ω_2	-0.1	-0.1638	(0.2637)	-0.0722	(0.0976)	-0.1541	(0.1586)	-0.0547	(0.0956)
ω_3	-0.02	-0.0796	(0.4246)	-0.0726	(0.1381)	-0.1021	(0.3821)	-0.1204	(0.2928)
Φ_{11}	0.98	0.9680	(0.0480)	0.9533	(0.0841)	0.9686	(0.0236)	0.9635	(0.0198)
Φ_{22}	0.98	0.9631	(0.0382)	0.9536	(0.0979)	0.9655	(0.0224)	0.9750	(0.0174)
Φ_{33}	0.98	0.9451	(0.1057)	0.9579	(0.0697)	0.9405	(0.1039)	0.9540	(0.1155)
Φ_{12}	0	0.0026	(0.0398)	-0.0222	(0.0679)	0.0007	(0.0206)	-0.0104	(0.0154)
Φ_{13}	0	0.0050	(0.0630)	-0.0067	(0.0554)	0.0062	(0.0470)	0.0013	(0.0321)
Φ_{23}	0	-0.0012	(0.0612)	-0.0098	(0.0773)	-0.0041	(0.0442)	0.0025	(0.0327)
σ_β	0.3					0.3042	(0.0150)		
ρ_β		2.0194	(0.2651)	6.0372	(0.9980)			10.1391	(0.9873)
$l(\hat{\psi})$	-2683.2	-2651.9		-2676.4		-2671.6		-2725.7	
MSFE	0.3820	0.3857		0.3909		0.3853		0.3978	

Note: The table reports the estimation results and forecasting performance of the ProPar filter, the DCS filter, the Kalman filter and the observation-driven variant of the Kalman filter (OD) in the three-factor Nelson-Siegel framework. The data is generated using a state-space model. For each static parameter in the model, the true value, the estimates and its standard deviation (between brackets) are given. Also, the log-likelihood value on the train-sample, $l(\hat{\psi})$, and the mean squared forecast error (MSFE) on the test data are displayed for each filtering method. The 'true' log-likelihood and MSFE are those of the Kalman filter where the 'true' parameter values are used. The estimates, log-likelihood values and MSFEs are obtained by taking the average over 250 iterations, and the standard deviation is the standard deviation over 250 iterations. The 250 simulated data sets each consist of $T = 1000$ observations and a train/test-split of 80/20 is used.

Table 8: Simulation results of various filtering methods in the four-factor Nelson-Siegel model using state-space generated data.

	Truth	ProPar filter		DCS filter		Kalman filter		Kalman filter (OD)	
Σ_{11}	0.2	0.4385	(0.0868)	0.4551	(0.0369)	0.1947	(0.0176)	0.3571	(0.0432)
Σ_{22}	0.2	0.4123	(0.0664)	0.4485	(0.0415)	0.1949	(0.0176)	0.3468	(0.0443)
Σ_{33}	0.2	0.4094	(0.0623)	0.4496	(0.0388)	0.1961	(0.0174)	0.3472	(0.0461)
Σ_{44}	0.2	0.4127	(0.0716)	0.4369	(0.0383)	0.2012	(0.0157)	0.3538	(0.0441)
Σ_{12}	0	0.2075	(0.0519)	0.2408	(0.0375)	-0.0057	(0.0141)	0.1407	(0.0367)
Σ_{13}	0	0.1726	(0.0445)	0.2185	(0.0377)	-0.0052	(0.0132)	0.1242	(0.0388)
Σ_{14}	0	0.1351	(0.0435)	0.1813	(0.0341)	-0.0052	(0.0132)	0.1020	(0.0378)
Σ_{23}	0	0.1945	(0.0432)	0.2397	(0.0382)	-0.0045	(0.0131)	0.1396	(0.0400)
Σ_{24}	0	0.1525	(0.0471)	0.2000	(0.0359)	-0.0035	(0.0122)	0.1138	(0.0390)
Σ_{34}	0	0.1776	(0.0490)	0.2213	(0.0351)	-0.0026	(0.0130)	0.1306	(0.0376)
ω_1	0.1	0.1477	(0.1260)	0.0778	(0.4230)	0.1882	(0.2173)	0.1088	(0.1251)
ω_2	-0.1	-0.1545	(0.1696)	-0.1941	(0.3758)	-0.2620	(0.3523)	-0.1241	(0.1515)
ω_3	-0.02	0.0156	(0.3717)	0.0504	(0.5733)	-0.0417	(0.5633)	-0.0765	(0.3069)
ω_4	-0.06	-0.1463	(0.2566)	-0.4170	(0.7373)	-0.2012	(0.4340)	-0.0865	(0.1778)
Φ_{11}	0.98	0.9700	(0.0320)	0.9027	(0.1423)	0.9580	(0.0451)	0.9694	(0.0277)
Φ_{22}	0.98	0.9597	(0.0461)	0.9301	(0.0892)	0.9358	(0.0704)	0.9690	(0.0364)
Φ_{33}	0.98	0.9009	(0.1932)	0.8767	(0.1571)	0.8712	(0.1723)	0.9429	(0.1427)
Φ_{44}	0.98	0.9506	(0.0819)	0.7994	(0.2894)	0.9166	(0.1232)	0.9620	(0.0551)
Φ_{12}	0	0.0033	(0.0213)	-0.0459	(0.0908)	0.0004	(0.0300)	-0.0042	(0.0154)
Φ_{13}	0	0.0070	(0.0597)	0.0282	(0.1325)	0.0077	(0.0554)	0.0015	(0.0440)
Φ_{14}	0	0.0052	(0.0238)	-0.0606	(0.1053)	0.0008	(0.0319)	-0.0089	(0.0185)
Φ_{23}	0	-0.0120	(0.0734)	0.0120	(0.1137)	0.0045	(0.1019)	0.0092	(0.0534)
Φ_{24}	0	0.0046	(0.0301)	-0.0493	(0.0790)	0.0008	(0.0560)	-0.0135	(0.0390)
Φ_{34}	0	0.0218	(0.0973)	0.0787	(0.1683)	0.0256	(0.0940)	0.0125	(0.0663)
σ_β	0.3					0.3108	(0.0176)		
σ_λ	0.2					0.1841	(0.0277)		
ρ_β		4.9744	(12.6103)	6.6160	(5.7863)			14.1730	(9.2888)
ρ_λ		21.6095	(15.5668)	17.5815	(10.4315)			27.4077	(8.3792)
$l(\hat{\psi})$	-2812.4	-2914.0		-2815.3		-2794.3		-2897.0	
MSFE	0.4371	0.4912		0.4749		0.4424		0.5118	

Note: The table reports the estimation results and forecasting performance of the ProPar filter, the DCS filter, the Kalman filter and the observation-driven variant of the Kalman filter (OD) in the four-factor Nelson-Siegel framework. The data is generated using a state-space model. For each static parameter in the model, the true value, the estimates and its standard deviation (between brackets) are given. Also, the log-likelihood value on the train-sample, $l(\hat{\psi})$, and the mean squared forecast error (MSFE) on the test data are displayed for each filtering method. The 'true' log-likelihood and MSFE are those of the Kalman filter where the 'true' parameter values are used. The estimates, log-likelihood values and MSFEs are obtained by taking the average over 250 iterations, and the standard deviation is the standard deviation over 250 iterations. The 250 simulated data sets each consist of $T = 1000$ observations and a train/test-split of 80/20 is used.

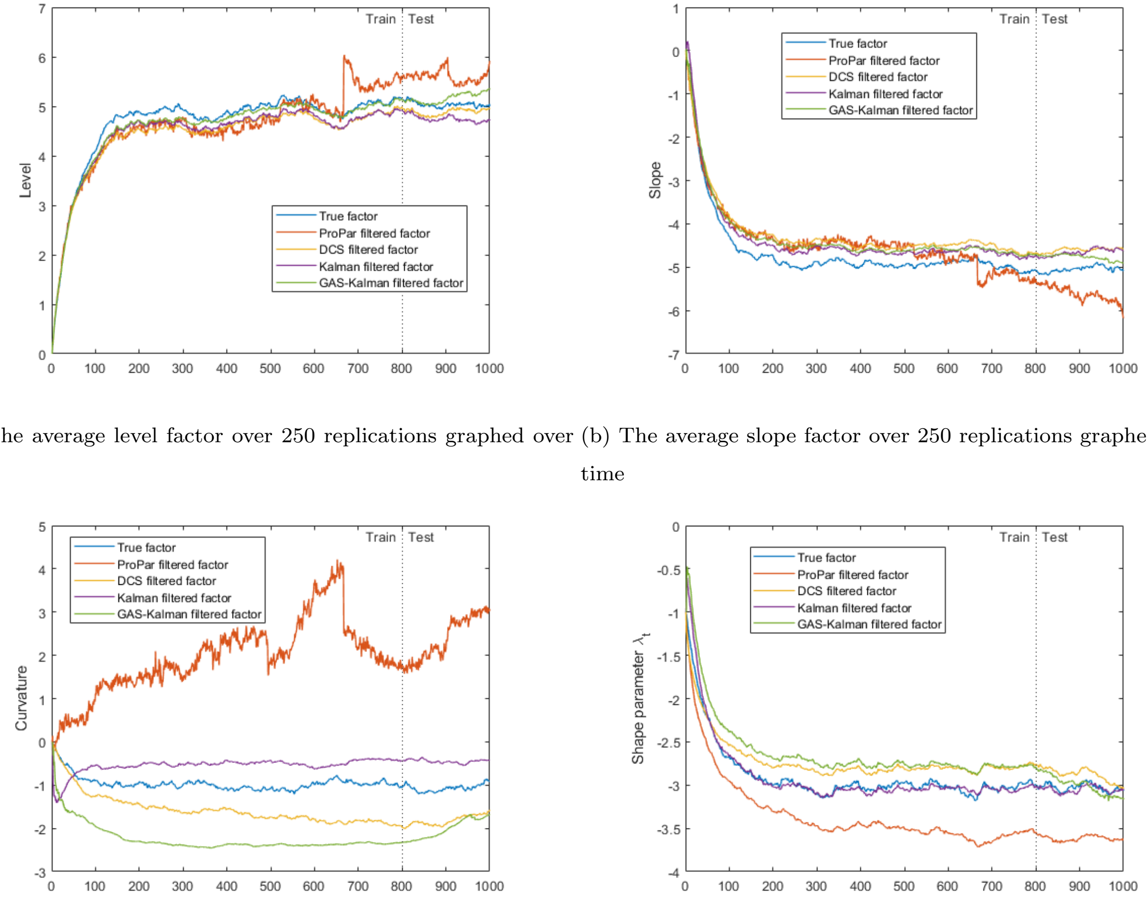


(a) The average level factor over 250 replications graphed over time

(b) The average slope factor over 250 replications graphed over time

(c) The average curvature factor over 250 replications graphed over time

Figure 3: **The replication-averaged level, slope, and curvature factors over time in the NS3F model with state-space-dynamics.** This figure shows the filtered level (panel a), slope (panel b), and curvature (panel c) factors in the NS3F model with state-space-dynamics averaged over the 250 replications. Next to the true simulated factor values, the figure shows the filtered factor values of the ProPar filter, the DCS filter, the Kalman filter and the GAS-Kalman filter.



(a) The average level factor over 250 replications graphed over time

(b) The average slope factor over 250 replications graphed over time

(c) The average curvature factor over 250 replications graphed over time

(d) The average shape parameter over 250 replications graphed over time

Figure 4: **The replication-averaged level, slope, and curvature factors over time in the NS4F model with state-space-dynamics.** This figure shows the filtered level (panel a), slope (panel b), and curvature (panel c) factors, and the logarithm of the shape parameter (panel d) in the NS4F model with state-space-dynamics averaged over the 250 replications. Next to the true simulated factor values, the figure shows the filtered factor values of the ProPar filter, the DCS filter, the Kalman filter and the GAS-Kalman filter.

Table 9: Filtering performance in the NS3F model with state-space-dynamics in terms of MSFE statistics.

	ProPar filter		DCS filter		Kalman filter		GAS-Kalman filter		VAR(1)		Naive forecast	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	1.639	3.382	1.442	0.641	1.290	0.616	1.852	0.986	1.531	0.571	3.756	0.421
Slope factor	1.544	3.413	1.657	0.592	0.998	0.241	2.351	1.139	1.591	0.382	2.238	0.281
Curvature factor	8.535	17.636	11.892	8.817	7.069	4.840	12.056	9.727	14.240	9.546	29.845	3.108

Note: The table reports the filtering performance, in terms of average mean squared forecast error over 250 replications (*Mean MSFE*) and standard deviation of the mean squared forecast errors from the 250 replications (*Stdev*) of the time-varying factors in the NS3F model. The NS3F model specification includes a state-space structure for the dynamics of the factors. The filters that are considered are the ProPar filter, the dynamic conditional score (*DCS*) filter, the Kalman filter, a version of the Kalman filter that assumes that the dynamics of the factors are GAS-specified (*GAS-Kalman*), the first-order vector autoregressive (*VAR[1]*) model and the naive forecast model. The mean squared errors at each repetition are calculated over the test sample which consists of the last 20% of the simulated data.

Table 10: Filtering performance in the NS4F model with state-space-dynamics in terms of MSFE statistics.

	ProPar filter		DCS filter		Kalman filter	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	67.721	700.699	3.338	18.734	1.094	1.072
Slope factor	77.837	746.710	3.292	15.226	1.252	1.086
Curvature factor	178.729	1354.838	14.783	51.997	6.533	11.240
Shape parameter log λ_t	10.465	87.865	2.012	13.395	0.568	0.523

	GAS-Kalman filter		VAR(1)		Naive forecast	
	Mean MSFE	Stdev	Mean MSFE	Stdev	Mean MSFE	Stdev
Level factor	5.092	29.315	12.491	6.967	168.640	105.416
Slope factor	5.534	25.084	20.516	9.993	288.753	122.624
Curvature factor	21.999	76.427	19.654	15.231	551.239	204.266
Shape parameter log λ_t	5.256	44.681	1.263	0.880	1.995	1.028

Note: The table reports the filtering performance, in terms of average mean squared forecast error over 250 replications (*Mean MSFE*) and standard deviation of the mean squared forecast errors from the 250 replications (*Stdev*) of the time-varying factors in the NS4F model. The NS4F model specification includes a state-space structure for the dynamics of the factors. The filters that are considered are the ProPar filter, the dynamic conditional score (*DCS*) filter, the Kalman filter, a version of the Kalman filter that assumes that the dynamics of the factors are GAS-specified (*GAS-Kalman*), the first-order vector autoregressive (*VAR[1]*) model and the naive forecast model. The mean squared errors at each repetition are calculated over the test sample which consists of the last 20% of the simulated data.

5 Empirical study

To see whether the ProPar filter in the Nelson-Siegel framework is suitable for forecasting yield curves, we apply the method to U.S. yield data. After discussing the yield data in Section 5.1, we present our findings in Section 5.2.

5.1 Yield data

The yield data that is used to perform the empirical study is provided by Liu and Wu (2021) and consists of U.S. yields derived from government bonds. The sample ranges from November 1971 to October 2021 and covers 600 monthly observations. It contains yields with a maturity of 1, 2, ..., 360 months. However, for this research, we only consider a subset of $N = 16$ maturities as these contain sufficient information to construct yield curves. The chosen maturities range from 3 to 180 months. Figure 5 displays a short-, medium- and long-term yield over time. Some remarkable periods in time are the early 1980s recession in which annual inflation peaked at almost 15%, the bursting of the dot-com bubble around 2000, and the great recession in 2008.

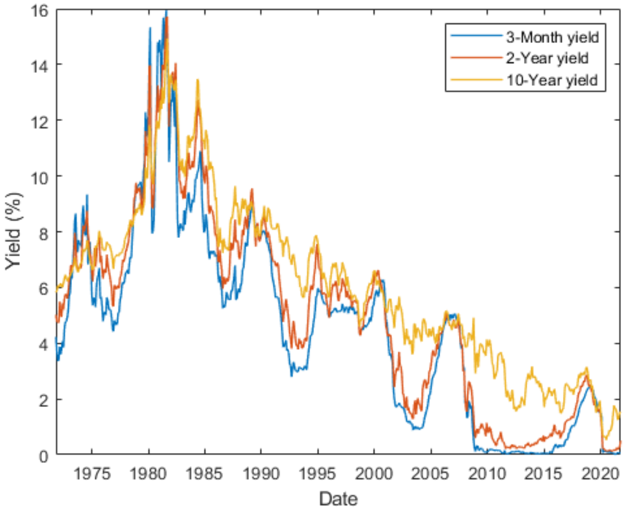


Figure 5: **Short-, medium-, and long-term U.S. yields over time.** This figure shows short-, medium-, and long-term U.S. yields that correspond to a maturity of 6 months, 2 years and 10 years respectively, over the period of November 1971 to October 2021.

Table 11 presents descriptive statistics of the data, which reveal that longer-term yields tend to be higher than shorter-term yields. This reflects the expectation of higher interest rates in the future and implies the generally positive slope of the yield curve. From maturities of at least 9 months, we see that the volatility decreases by maturity. When one looks at the autocorrelation coefficients, one can see that the yield curves tend to display a certain degree of persistence over time. Also, yields for long maturities are more persistent than yields for shorter maturities. Changes in the yield curve typically occur gradually rather than abruptly, reflecting the gradual adjustment of market expectations and economic factors. Moreover, according to the cross-correlation matrix in Table 16 (Appendix A), yields for different maturities have high cross-correlations. These reflections are all in line with several commonly observed patterns and characteristics of the yield curve.

When one fits the Nelson-Siegel model to the yield curve at each point in time, one can find the factor values that result in the best fit. In the NS3F model, one needs to find an appropriate value for λ_t. The shape parameter determines the maturity at which the loading on the curvature factor achieves it maximum. For this purpose, it is common to use medium-term maturities. Similar to Diebold and Li (2006), we choose a maturity of 30 months. The λ_t value that maximizes the loading on the curvature factor at 30 months is 0.0609. Hence, we fix shape parameter λ_t in the three-factor

Table 11: Descriptive Statistics of the U.S. Yield Data

Maturity	Mean	Stdev	Min	Max	r_1	r_{12}	r_{60}
3	4.533	3.530	0.009	15.946	0.989	0.860	0.547
6	4.689	3.583	0.028	16.133	0.990	0.869	0.557
9	4.801	3.597	0.045	16.107	0.990	0.875	0.568
12	4.886	3.596	0.064	15.962	0.990	0.881	0.580
18	5.022	3.591	0.106	15.943	0.991	0.890	0.602
24	5.122	3.558	0.123	15.719	0.992	0.896	0.619
30	5.215	3.513	0.114	15.545	0.992	0.900	0.631
36	5.316	3.482	0.123	15.569	0.992	0.904	0.636
48	5.497	3.410	0.172	15.475	0.992	0.907	0.645
60	5.631	3.326	0.231	15.195	0.993	0.910	0.652
72	5.767	3.273	0.313	14.989	0.993	0.912	0.652
84	5.868	3.209	0.382	14.950	0.993	0.911	0.649
96	5.956	3.157	0.445	14.941	0.993	0.914	0.646
108	6.028	3.108	0.488	14.945	0.993	0.914	0.643
120	6.086	3.039	0.530	14.939	0.993	0.910	0.643
180	6.406	2.989	0.734	14.914	0.992	0.912	0.634

Note: The table reports the summary statistics for the U.S. zero-coupon bond rates for different maturities (denoted in months). The data ranges from November 1971 until October 2021, resulting in $T = 600$ monthly observations. For each of the $N = 16$ maturities, the mean, standard deviation (*Stdev*), minimum and maximum, and autocorrelations r_1, r_{12}, and r_{60} on 1, 12, and 60 month lags respectively are shown.

model to 0.0609. Note that we only impose this constraint in preliminary research to describe the level, slope and curvature factor. The λ_t estimate in the NS3F model is obtained as described in Section 3. Table 12 displays several statistics about these factors. Again, the level factor is reflected by the long-term yield as the slope and curvature loadings shrink to zero when maturity increases. Hence, it is not surprising that the level factor is generally positive. The slope factor is closely related to the short-term yield minus the long-term yield. Thus, its sign depends on the shape of the yield curve, but is generally negative as yield curves tend to be increasing. The curvature factor usually affects medium-term yields. In the NS3F model, it is clear that the factors are persistent. Especially the level factor which has an autocorrelation on its 12-month lag of 0.906. In the NS4F model, the factors are not highly persistent. This can be caused by the outliers that are present in the level, slope, and curvature factors as soon as the shape parameter changes over time. This may have consequences for the dynamic modeling of the factors, as persistence is a fundamental component in any time series model.

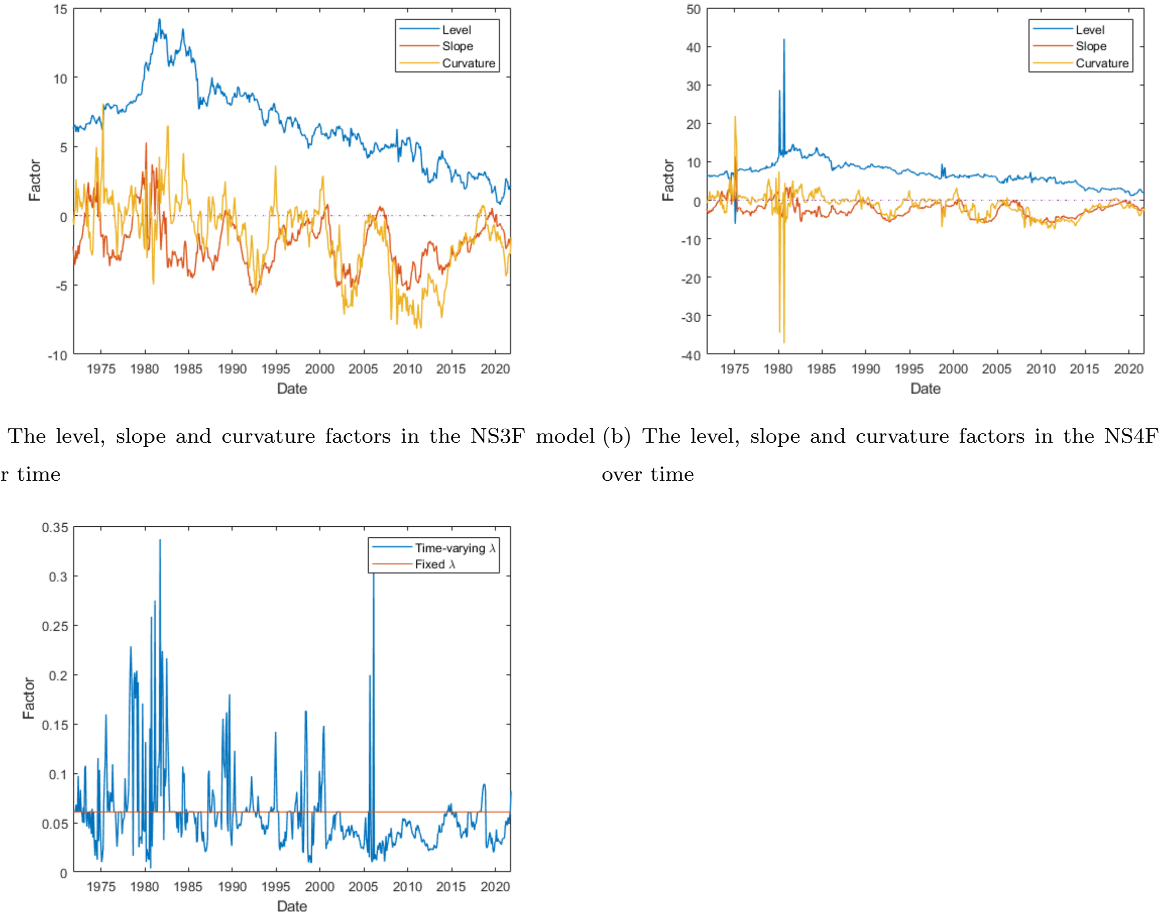
Table 12: Descriptive Statistics of the estimated factors in the three- and four-factor Nelson-Siegel model

NS3F model	Mean	Stdev	Min	Max	r_1	r_{12}	r_{60}
Level	6.571	2.860	0.794	14.223	0.992	0.906	0.606
Slope	-1.996	1.871	-5.561	5.275	0.956	0.521	-0.061
Curvature	-1.462	2.688	-8.157	8.091	0.931	0.631	0.230
NS4F model							
Level	6.913	3.279	-6.105	41.931	0.785	0.691	0.386
Slope	-2.457	2.401	-30.087	11.376	0.640	0.308	0.006
Curvature	-1.388	3.509	-37.147	21.902	0.496	0.289	0.027
$\log \lambda_t$	-3.011	0.575	-5.606	-1.089	0.532	0.158	0.059

Note: The table reports the summary statistics for the estimated factors in the three- and four-factor Nelson-Siegel models. The data used is U.S. yield data ranging from November 1971 until October 2021, resulting in $T = 600$ monthly observations. For each of the factors the mean, standard deviation (*Stdev*), minimum and maximum, and autocorrelations r_1, r_{12}, and r_{60} on 1, 12, and 60 month lags respectively are shown.

Figure 6 graphs the estimated factors over time. The level, slope, and curvature factors behave similar across the three- and four-factor models. At some points in time, when we allow the shape parameter of the factor loadings to be dynamic, the level, slope, and curvature factors contain outliers. At these time points, the yield curve takes on unusual shapes. The four time-varying factors take on extreme values in order to fit these shapes. For instance, in the period just after 1980, we detect two points in time at which the level, slope, and curvature take on unusual values in the NS4F model compared to the factor values from the NS3F model. When we look at the yields around that period in Figure 5, we see that the 10-year yield is lower than the 6-months and 2-year yields, indicating that the yield curve was not monotonically increasing as usual.

The relatively high values of λ_t in the NS4F model correspond to points in time where the yield curve was not monotonically increasing. During the periods where the yield curve was clearly increasing, for instance between 2010 and 2015 where according to Figure 5 the long-term yield is clearly higher than shorter-term yields, shape parameter λ_t is estimated to be lower and less volatile.



(a) The level, slope and curvature factors in the NS3F model over time

(b) The level, slope and curvature factors in the NS4F model over time

(c) The fixed shape parameter (NS3F model) versus the time-varying shape parameter (NS4F model) over time

Figure 6: **Factor values from the Nelson-Siegel three- and four-factor models over time.** This figure shows the estimated level, slope, and curvature factors in the NS3F model (panel a), the estimated level, slope, and curvature factors in the NS4F model (panel b), and the time-varying shape parameter λ_t (panel c). In the NS3F model, shape parameter λ_t is fixed at a value of 0.0609.

5.2 Findings

As the yield data contains $N = 16$ maturities the number of parameters, especially in Σ_ϵ, becomes large. To ensure computational convenience, we impose several constraints. First of all, we force the estimate of Σ_ϵ to be a scalar multiple of the covariance matrix of yields of the train sample, i.e. $\hat{\Sigma}_\epsilon = \hat{\sigma}_\epsilon Cov(y)$ where σ_ϵ is a scalar. Secondly, for the learning-rate matrix H_t, we use the inverse of the penalty matrices form Equation 12 (NS3F model) and Equation 17 (NS4F) using $k = 1$, implying we use a multiple of the predictive Fisher matrix as penalty matrix. However, to enhance convenience, we assume the mixed partial derivatives of β_t and $\log \lambda_t$ are zero such that the upper-right and lower-left block matrices of the predictive Fisher matrix are zero in the four-factor case. Hence, the learning-rate matrix in the four-factor model becomes

$$\begin{aligned} H_t &= \begin{bmatrix} \rho_\beta^{-1} I_3 & 0 \\ 0 & \rho_\lambda^{-1} \end{bmatrix} \mathcal{I}_{t|t-1}^{-1} \\ &= \begin{bmatrix} \rho_\beta^{-1} \left[\Lambda(\lambda_{t|t-1})' \Sigma_\epsilon^{-1} \Lambda(\lambda_{t|t-1})\right]^{-1} & 0 \\ 0 & \rho_\lambda^{-1} \left[\beta_{t|t-1}' \dot{\Lambda}(\lambda_{t|t-1})' \Sigma_\epsilon^{-1} \dot{\Lambda}(\lambda_{t|t-1}) \beta_{t|t-1}\right]^{-1} \end{bmatrix}, \end{aligned}$$

where $\dot{\Lambda}(\lambda)$ is the derivative of $\Lambda(\lambda)$ with respect to $\log \lambda$. Lastly, in the NS4F model, we only have to estimate ω_4, Φ_{44} and ρ_λ. The other parameter estimates equal the corresponding estimates from the NS3F model, except for the off-diagonal elements in the fourth row/column of Φ which are set to zero. Moreover, we take the logarithm of the estimate of λ from the NS3F model as the initial value for the $\log \lambda_0$ in the NS4F. In this way, the NS4F model can converge to the NS3F model (by setting $\hat{\omega}_4 = 0$, $\hat{\Phi}_{44} = 1$, and $\hat{\rho}_\lambda = \infty$), implying that the performance of the NS4F model is at least as good as the performance of the NS3F model.

Table 13: Estimation results and forecasting performance of the ProPar filtering method and various benchmark filtering methods on the three- and four-factor Nelson-Siegel model using U.S. yield data.

NS3F model	ProPar filter	DCS filter	Kalman filter	Kalman filter (OD)
λ	0.0439	0.0439	0.0403	0.0397
σ_ϵ	0.0208	0.0208	0.0015	0.0014
ω_1	0.1636	0.1636	0.1885	0.1895
ω_2	-0.1959	-0.1959	-0.1282	-0.2027
ω_3	-0.2189	-0.2189	-0.1914	-0.2845
Φ_{11}	0.9833	0.9833	0.9783	0.9808
Φ_{22}	0.9493	0.9493	0.9558	0.9534
Φ_{33}	0.8646	0.8646	0.9402	0.8408
Φ_{12}	0.0129	0.0129	0.0052	0.0138
Φ_{13}	0.0180	0.0180	0.0226	0.0259
Φ_{23}	0.0182	0.0182	0.0159	0.0126
ρ_β	-0.1157	0.8843		0.1275
σ_β			0.6153	
$l(\hat{\psi})$	-4100.5	-4100.5	3359.1	3222.1
MSFE	0.0313	0.0313	0.0319	0.0309

NS4F model				
ω_4	-0.0292	0.0063	-0.6709	0.0061
Φ_{44}	0.9916	1.0022	0.7867	1.0019
ρ_λ	301.1769	424.5833		1.8982E+12
σ_λ			0.3398	
$l(\hat{\psi})$	-4076.8	-4094.2	3967.0	3252.2
MSFE	0.0301	0.0302	0.0311	0.0305

Note: The table reports the estimation results and forecasting performance of the ProPar filter, DCS filter, Kalman filter, and the observation-driven version of the Kalman filter in the three- (NS3F) and four-factor (NS4F) Nelson-Siegel framework on U.S. yield data. The estimates for the static parameters are shown. Also, the log-likelihood value on the train sample, $l(\hat{\psi})$, and the mean squared forecast error (MSFE) on the test data are displayed for each filtering method. The U.S. yield data consists of $T = 600$ monthly observations (November 1971 until October 2021) and $N = 16$ different maturities per point in time (ranging from 3 months to 15 years). The train sample uses the first 80% of the data, and the remaining 20% is used for out-of-sample forecasting evaluation.

Table 13 shows the results of the NS3F and NS4F models using all filtering methods. As shown in Section 3.1, the update step of the ProPar filter can be rewritten to the update step of the DCS filter when the a scalar multiple of the predictive Fisher matrix is used as penalty matrix and the observation equation is linear in the factors. The two filters converge to the same filter and the linear transformation in the penalty scalar ρ_β makes their estimates differ by exactly one.

Also, we see that the log-likelihood on the train sample is extremely high for the Kalman filters. This is a result of the estimate for σ_ϵ, the scalar with which the covariance matrix of yields is multiplied to get the estimate for Σ_ϵ. As yield data is highly correlated (see the correlation matrix in Table 16), the covariance matrix of yields has a very low determinant, namely 2.385E-35. The Kalman and GAS-Kalman filter estimate σ_ϵ around 0.0015, resulting in $\hat{\Sigma}_\epsilon$ with a determinant of 1.566E-80. This near-zero determinant suggests multicollinearity, which is inevitably present in the highly correlated yield data. The multicollinearity problem is beyond the scope of this research, but the low determinant of $\hat{\Sigma}_\epsilon$ explains why the log-likelihood value is this high. We can therefore not draw any valid conclusions on which filter has the best model fit.

The estimates of ρ_λ are of significantly higher magnitude than the estimates of ρ_β, indicating that the time-varying sequences β_t and $\log \lambda_t$ should be penalized differently to get the best model fit. One could also look into a filter with different penalty scalars for the level, slope, and curvature factors, which is beyond the scope of this research.

The GAS-Kalman filter provides the most accurate out-of-sample yield forecasts in the NS3F model. The Kalman filter generates the highest out-of-sample mean squared forecast error among all filters. When we let $\log \lambda_t$ vary over time, we see that all model fits and forecasting performances increase. The GAS-Kalman filter increases the least, which is also indicated by its parameter estimates for ω_4, Φ_{44} and ρ_λ which are closest to the settings that make the NS4F model converge to the NS3F model ($\omega_4 = 0$, $\Phi_{44} = 1$, and $\rho_\lambda = \infty$). The ProPar filter beats the DCS filter by a tiny margin in terms of forecasting and has an MSFE that is 3.2% smaller than the MSFE that the Kalman filter can attain. The gain from adding the shape parameter as a fourth factor is not as large as one would anticipate after reading the work of Koopman et al. (2010). However, as our NS4F model is highly restrictive, implying that the estimated level, slope, and curvature factors are the same in both models, the gain is marginal.

Table 14 reports the MSFE statistics for the simple benchmark models. The filtering methods from Table 13 are able to beat all the simple benchmarks, except for the naive forecasting model in a four-factor framework. This finding suggests that the factors are sufficiently persistent in order to be explained by this simple model. Moreover, the naive forecasting model, which includes fewer parameters and assumptions, is less prone to overfitting and therefore leading to better generalization to new, unseen data. Nevertheless, the difference in MSFE between the ProPar filter and the naive forecasting model is small (4.2%).

Table 14: Yield forecasting performance of several benchmark models.

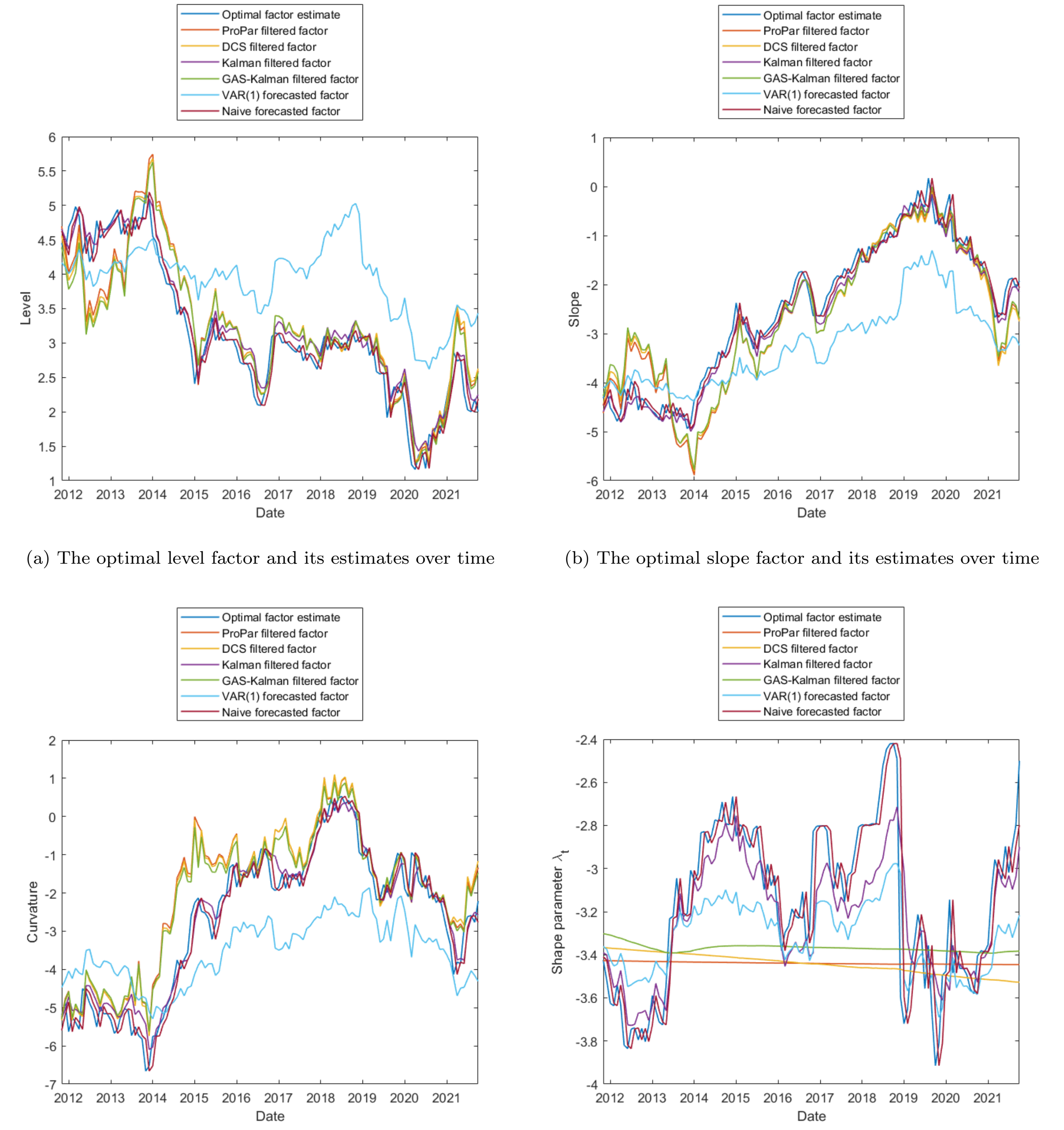
	MSFE
VAR[1] β_t	0.0358
VAR[1] on θ_t	0.1139
Naive β_t forecast	0.0351
Naive θ_t forecast	0.0289

Note: The table reports the forecasting performance, in terms of the mean squared forecast error (MSFE) of out-of-sample yield forecasts. The models that are considered are the first-order vector autoregressive (*VAR[1]*) model and a naive forecasting model on the factors in the three- and four-factor Nelson-Siegel framework. The factors in the three-factor framework are denoted by β_t, while the factors in the four-factor framework are denoted by θ_t. The U.S. yield data consists of $T = 600$ monthly observations (November 1971 until October 2021) and $N = 16$ different maturities per point in time (ranging from 3 months to 15 years). The training sample uses the first 80% of the data, and the remaining 20% is used for out-of-sample forecasting evaluation.

We have to judge our filtering methods not only on the yield forecasts, but also on their ability to predict the state vector. The optimal factor values, which we derived in Section 5.1 via cross-sectional regressions, are graphed in Figure 7 along with their predicted values from the considered methods. One can easily see that the Kalman filter and the naive forecasting model consistently provide the most accurate state predictions on the test sample. While the ProPar, DCS and GAS-Kalman filters are able to predict reasonably close to the optimal pattern of the level, slope, and curvature factors, the predictions of the shape parameters seem almost constant. This can be explained by the relatively high estimates of penalties on the shape parameter ρ_λ from Table 13.

These views are confirmed by the mean squared forecast error statistics of the factor values, given in Table 15. The MSFE statistics from this table indicate the difference between the method's state prediction and the optimal state that would result in the best fit.

In short, predicting the one-step-ahead state can best be done in a naive way, by setting the prediction equal to today's fitted values. This method also generates the most accurate out-of-sample yield forecasts. However, given the highly restricted version of the four-factor Nelson-Siegel framework that we use for the more advanced filters in this empirical study, the results of the Kalman and ProPar filters are promising. The Kalman filter is able to accurately predict the state vector, and the ProPar filter generates precise out-of-sample yield forecasts.



(a) The optimal level factor and its estimates over time

(b) The optimal slope factor and its estimates over time

(c) The optimal curvature factor and its estimates over time

(d) The optimal shape parameter and its estimates over time

Figure 7: **Factor estimates from the four-factor Nelson-Siegel model over time.** This figure shows the optimal level (panel a), slope (panel b), and curvature (panel c) factors and the logarithm of the time-varying shape parameter $\log \lambda_t$ (panel d), along with six factor estimates. The estimates are obtained using the ProPar filter, the DCS filter, the Kalman filter, the GAS-Kalman filter, the VAR(1) model and the naive forecasting model. The methods are trained on monthly U.S. yield data ranging from November 1971 to October 2011. The test sample ranges from November 2011 to October 2021, which is graphed in the figure. The optimal factor values are obtained via cross-sectional least squares.

Table 15: Out-of-sample state filtering performance.

	MSFE of ProPar filter	MSFE of DCS filter	MSFE of Kalman filter	MSFE of GAS-Kalman filter	MSFE of VAR(1)	MSFE of naive forecast
Level factor	0.224	0.245	0.068	0.243	1.217	0.054
Slope factor	0.327	0.351	0.071	0.343	1.080	0.064
Curvature factor	1.034	0.965	0.192	0.840	2.347	0.169
Shape parameter $\log \lambda_t$	0.209	0.218	0.043	0.178	0.084	0.028

Note: The table reports the filtering performance, in terms of the mean squared forecast error (MSFE) from a method's factor estimate to the optimal factor value. The methods that are considered are the ProPar filter, the dynamic conditional score (*DCS*) filter, the Kalman filter, the GAS-Kalman filter, a first-order vector autoregressive (*VAR[1]*) model and a naive forecasting model. All methods are applied to the four-factor Nelson-Siegel framework. The U.S. yield data that is used consists of $T = 600$ monthly observations (November 1971 until October 2021) and $N = 16$ different maturities per point in time (ranging from 3 months to 15 years). The training sample uses the first 80% of the data, and the remaining 20% is used for calculating the out-of-sample filtering performances which are given in the table.

6 Conclusion

This thesis aims to find an answer to the questions of whether the ProPar filter in a dynamic Nelson-Siegel framework is suitable for forecasting yield curves, and whether the ProPar filter can outperform several widely-used benchmark filters in this framework.

The simulation experiment demonstrates that the ProPar filter, the dynamic conditional score (DCS) filter, the Kalman filter, and the GAS-Kalman filter are all capable of estimating the parameters in the Nelson-Siegel framework reasonably well for at least one of the considered model specifications. Furthermore, these filters exhibit a good fit to the data, closely approximating the correctly specified filter with the true parameter values. This holds true for both the three- and four-factor models, as well as for both the state-space and generalized autoregressive score data specifications.

In terms of out-of-sample forecasting, these advanced filtering methods outperform simpler benchmark models such as the first-order vector autoregressive (VAR[1]) model and the naive forecasting model. Notably, the Kalman filter excels in out-of-sample forecasting, demonstrating its ability to effectively capture underlying patterns that generalize well to unseen data in the simulation context. Surprisingly, this advantage persists even when the Kalman filter has a higher degree of misspecification compared to other filters. The simulation setup reveals that the specification of factor dynamics plays a significant role in the filtering performance of these methods. Specifically, the DCS filter demonstrates the highest filtering accuracy in the GAS-specified factors, while the

Kalman filter is superior in filtering the factors in the state-space setting.

Despite the ability of the Kalman filter to predict yields most accurately in the simulation experiment, the empirical study using U.S. yield data suggests that the ProPar filter offers slightly improved one-month-ahead yield forecasts compared to other advanced filtering methods. The ProPar filter appears to have the ability to distill the underlying driving factors of yield curves effectively. It's noteworthy that although the Kalman filter seems to predict the state vector most accurately, it exhibits the highest mean squared yield forecast error among the four filters. This indicates that the true factors in real yield data may be described by a Gaussian process, but effectively filtering the state vector does not guarantee more accurate yield forecasts.

Surprisingly, the naive forecasting model, which predicts the factors one month ahead based on today's fitted values, marginally outperforms the ProPar filter in out-of-sample forecasting. This result not only confirms the persistence of the factors but also serves as a cautionary note about overfitting when dealing with yield data. The outstanding out-of-sample forecasting performance of this simple model highlights that a model with fewer assumptions and parameters is less susceptible to overfitting and, consequently, leads to better generalization to new, unseen data.

Besides the theoretical implications of our results, we stress their practical benefits. Namely, the ProPar filter is able to accurately forecast the yield curve in times of crisis. Examples of such events/periods are:

- On June 23, 2016, the United Kingdom's Brexit referendum took place. The unexpected result, where the majority voted to leave the European Union, led to market uncertainty and a flight to safety. The yield curve dropped on average by 26.8%. It is interesting to note that the ProPar filter kept the MSFE at a level of 0.121, while the Kalman filter has an average squared forecast error of 0.165. The DCS filter and the GAS-Kalman filter did a reasonable job with an MSFE of 0.120 and 0.123 respectively. Surprisingly, the naive forecasting model has a mean squared error loss of only 0.105 after this event.

- On July 31, 2019, The Federal Open Market Committee (FOMC) cut the federal funds rate by 0.25% to a target range of 2.00-2.25%. This marked the first rate cut since the financial crisis. This event caused the yield curve to drop by 19.1%. The Kalman filter forecasted the yield curve with a mean squared error of 0.185, whereas the ProPar filter, the DCS filter, the GAS-Kalman filter and the naive forecasting model provided more accurate forecasts leading to an MSFE of 0.164, 0.163, 0.170, and 0.151 respectively.

- The global COVID-19 pandemic led to a significant economic downturn. The Fed responded with aggressive monetary policy, cutting rates and implementing quantitative easing measures, impacting short and long-term yields. From December 2019, when the virus is believed to have originated, to March 2020, when the virus was officially declared to be a pandemic, the yield curve collapsed by 78.4%. The ProPar filter, the DCS filter, and the GAS-Kalman filter again did a reasonable job in this volatile period, forecasting the yield curve with a mean squared error of 0.172, 0.171, and 0.180 respectively. Again, the Kalman filter stayed behind with an MSFE of 0.213. The naive forecasting model, which generally provides the most accurate out-of-sample yield forecasts, has a mean squared forecast error of 0.254 during this period.

Upon examining the findings from both the simulation experiment and the empirical study, it becomes evident that implementing proximal parameter updates within a score-driven Nelson-Siegel framework yields a viable model for forecasting the term structure of interest rates. Furthermore, the ProPar filter outperforms other advanced filtering methods in predicting the yield curves, albeit with relatively minor distinctions. The fact that the naive forecasting model provides slightly more accurate out-of-sample yield forecasts indicates that overfitting is something we need to be cautious about when working with yield data. However, given the highly restrictive Nelson-Siegel version upon which the ProPar filter is applied, the application of proximal parameter updates in a Nelson-Siegel framework is promising.

The findings of this research can be helpful in a lot of fields, for instance in the field of finance and investments. Accurate yield curve forecasting helps investors, financial institutions, and portfolio managers make informed decisions regarding bond investments, interest rate expectations, and asset allocation strategies. Minor improvements in yield predictions can already lead to a great deal of profit.

Moving forward, there are several promising directions for future research in this domain. As Koopman et al. (2010) already show using the Kalman filter, allowing the volatility to vary over time can be beneficial when forecasting the yield curve. It can be interesting to see how the ProPar filter performs in that setup. Furthermore, one can allow for a time-varying mean (often referred to as shifting end-points) using the ProPar filter. This thesis has limited the use of the ProPar filter to the Nelson-Siegel model, but there are already numerous extensions of this model, such as the extension from Svensson (1994), that can yield an interesting application for the ProPar filter. Lastly, being aware of potential multicollinearity issues and employing appropriate techniques to handle them is crucial when using the Nelson-Siegel model in yield curve modeling. Although the ProPar filter could benefit from adjustments to prevent overfitting and multicollinearity, our approach opens the door to a promising new field of research.

References

Alvarez, C., Cademartori, D., Galea, M., Gonzalez, O., and Romo, C. (1992). Harvey's model for forecasting economic growth: An exploratory study in Chile. Working paper, Catholic University of Valparaiso.

Anderson, B. D. and Moore, J. B. (2012). *Optimal filtering*. Courier Corporation.

BIS (2005). Zero-coupon yield curves: Technical documentation. *Bank for International Settlements, Basel*, Volume 25.

Campbell, J. Y. (1987). Stock returns and the term structure. *Journal of Financial Economics*, 18(2):373–400.

Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. (1985). An intertemporal general equilibrium model of asset prices. *Econometrica: Journal of the Econometric Society*, 53(2):363–384.

Diebold, F. X. and Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):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(1-2):309–338.

Duffee, G. R. (2002). Term premia and interest rate forecasts in affine models. *The Journal of Finance*, 57(1):405–443.

Egorov, A. V., Hong, Y., and Li, H. (2006). Validating forecasts of the joint probability density of bond yields: Can affine models beat random walk? *Journal of Econometrics*, 135(1-2):255–284.

Fama, E. F. and French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics*, 25(1):23–50.

Harvey, C. R. (1989). Forecasting economic growth with the bond and stock markets. *Financial Analysts Journal*, 45(5):38–45.

Harvey, C. R. (1991a). Interest rate based forecasts of German economic growth. *Weltwirtschaftliches Archiv*, 127:701–718.

Harvey, C. R. (1991b). Les taux d'interet et la croissance economique en France. *Analyse Financiere*, 86:97–103.

Harvey, C. R. (1991c). The term structure and world economic growth. *Journal of Fixed Income*, 1(1):7–19.

Harvey, C. R. (1993). The yield curve, stock returns and the prediction of Canadian economic growth. Working paper, Duke University.

Harvey, C. R., Kirby, C., and Kaul, S. (1992). La capacita previsiva della struttura per scadenza dei tassi d'interesse Italiani in relazione alla crescita economica reale. Working paper, Gruppo IMI.

Heath, D., Jarrow, R., and Morton, A. (1992). Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica: Journal of the Econometric Society*, 60(1):77–105.

Hull, J. and White, A. (1990). Valuing derivative securities using the explicit finite difference method. *Journal of Financial and Quantitative Analysis*, 25(1):87–100.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Koopman, S. J., Mallee, M. I., and Van der Wel, M. (2010). Analyzing the term structure of interest rates using the dynamic Nelson–Siegel model with time-varying parameters. *Journal of Business & Economic Statistics*, 28(3):329–343.

Lange, R.-J., van Os, B., and van Dijk, D. J. (2022). Robust observation-driven models using proximal-parameter updates. *SSRN Electronic Journal*.

Liu, Y. and Wu, J. C. (2021). Reconstructing the yield curve. *Journal of Financial Economics*, 142(3):1395–1425.

Morrison, D. (1992). The currency markets: How much volatility ahead? *Goldman Sachs, London.*

Nelson, C. R. and Siegel, A. F. (1987). Parsimonious modeling of yield curves. *Journal of Business*, 60(4):473–489.

Quaedvlieg, R. and Schotman, P. (2022). Hedging long-term liabilities. *Journal of Financial Econometrics*, 20(3):505–538.

Svensson, L. E. (1994). Estimating and interpreting forward interest rates: Sweden 1992-1994. Working Paper 4871, National Bureau of Economic Research.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5(2):177–188.

A Correlation

Table 16: Correlation matrix

	$y(3)$	$y(6)$	$y(9)$	$y(12)$	$y(18)$	$y(24)$	$y(30)$	$y(36)$	$y(48)$	$y(60)$	$y(72)$	$y(84)$	$y(96)$	$y(108)$	$y(120)$	$y(180)$
$y(3)$	1.000	0.999	0.997	0.994	0.988	0.983	0.977	0.972	0.962	0.954	0.946	0.939	0.933	0.927	0.923	0.901
$y(6)$		1.000	0.999	0.998	0.993	0.988	0.983	0.979	0.969	0.962	0.954	0.948	0.942	0.936	0.931	0.909
$y(9)$			1.000	0.999	0.996	0.993	0.988	0.984	0.976	0.968	0.961	0.955	0.949	0.943	0.939	0.917
$y(12)$				1.000	0.998	0.996	0.992	0.988	0.981	0.974	0.967	0.961	0.956	0.950	0.946	0.924
$y(18)$					1.000	0.999	0.997	0.995	0.989	0.984	0.978	0.973	0.968	0.963	0.958	0.938
$y(24)$						1.000	0.999	0.998	0.994	0.990	0.985	0.980	0.976	0.971	0.967	0.948
$y(30)$							1.000	1.000	0.997	0.994	0.990	0.986	0.982	0.978	0.974	0.956
$y(36)$								1.000	0.999	0.996	0.993	0.990	0.986	0.983	0.979	0.962
$y(48)$									1.000	0.999	0.997	0.995	0.992	0.990	0.987	0.972
$y(60)$										1.000	0.999	0.998	0.996	0.994	0.991	0.979
$y(72)$											1.000	0.999	0.998	0.997	0.995	0.984
$y(84)$												1.000	1.000	0.999	0.997	0.988
$y(96)$													1.000	1.000	0.999	0.991
$y(108)$														1.000	0.999	0.994
$y(120)$															1.000	0.996
$y(180)$																1.000

Note: The table reports the correlation matrix of the 16 zero-coupon bond rates, where $y(\tau)$ denotes the yield with maturity τ in months.

B Mathematical derivations

B.1 Score vectors

Both the ProPar filter as well as its explicit version make use of the score vector of the observation density in order to update the factors. In the linear NS3F model from Equation 1, the log-density of observation y_t is given by

$$\log p(y_t|\beta_t) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_\epsilon| - \frac{1}{2}(y_t - \Lambda(\lambda)\beta_t)'\Sigma_\epsilon^{-1}(y_t - \Lambda(\lambda)\beta_t), \tag{20}$$

and has a corresponding score vector denoted by

$$\nabla(y_t|\beta_t) = \frac{\partial}{\partial\beta_t}\log p(y_t|\beta_t) = \Lambda(\lambda)'\Sigma_\epsilon^{-1}\big(y_t - \Lambda(\lambda)\beta_t\big). \tag{21}$$

In the NS4F model, the observation equation (Equation 4) becomes nonlinear in the factors. All dynamic factors are captured in $\theta_t' = (\beta_t', \log\lambda_t)$. The log-density of observation y_t becomes

$$\log p(y_t|\theta_t) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_\epsilon| - \frac{1}{2}(y_t - \Lambda(\lambda_t)\beta_t)'\Sigma_\epsilon^{-1}(y_t - \Lambda(\lambda_t)\beta_t), \tag{22}$$

with corresponding score vector

$$\nabla(y_t|\theta_t) = \begin{bmatrix} \frac{\partial}{\partial\beta_t}\log p(y_t|\theta_t) \\ \frac{\partial}{\partial\log\lambda_t}\log p(y_t|\theta_t) \end{bmatrix} = \begin{bmatrix} \Lambda(\lambda_t)'\Sigma_\epsilon^{-1}\big(y_t - \Lambda(\lambda_t)\beta_t\big) \\ \big(y_t - \Lambda(\lambda_t)\beta_t\big)'\Sigma_\epsilon^{-1}\dot{\Lambda}(\lambda_t)\beta_t \end{bmatrix}, \tag{23}$$

where $\dot{\Lambda}(\lambda) = \frac{\partial \Lambda(\lambda)}{\partial \log \lambda}$.

B.2 Hessian matrices

To determine the validity of the use of the ProPar filter, we evaluate the concavity of the logarithmic observation density. The NS3F model with observation log-density given in Equation 20, has Hessian matrix

$$\mathcal{H}(y_t|\beta_t) = \frac{\partial^2}{\partial \beta_t \partial \beta_t'} \log p(y_t|\beta_t) = -\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda). \tag{24}$$

This Hessian is negative definite as the function

$$f(x) = x' \Big(-\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda) \Big) x$$

achieves its unique maximum at

$$\nabla_x f(x) = -2\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda) x = \mathbf{0} \iff x = \mathbf{0}$$

as the columns of $\Lambda(\lambda)$ are all strictly positive column vectors and Σ_ϵ (and thus Σ_ϵ^{-1}) is a positive definite matrix by definition. Hence,

$$\begin{aligned} x' \Big(-\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda) \Big) x = f(x) \leq f(\mathbf{0}) = 0, && \forall x \in \mathbb{R}^3, \\ x' \Big(-\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda) \Big) x < 0, && \forall x \in \mathbb{R}^3 \setminus \{\mathbf{0}\}, \end{aligned}$$

which is a sufficient condition for $-\Lambda(\lambda)' \Sigma_\epsilon^{-1} \Lambda(\lambda)$ to be negative definite. The negative definiteness of the Hessian matrix implies that the log-likelihood is concave in β, which suggests the ProPar filter method to be appropriate for the NS3F model.

In the nonlinear NS4F model, the observation log-density is denoted in Equation 22. Its Hessian matrix is given by

$$\mathcal{H}(y_t|\theta_t) = \frac{\partial^2}{\partial \theta_t \partial \theta_t'} \log p(y_t|\theta_t) = \begin{bmatrix} \frac{\partial^2}{\partial \beta_t \partial \beta_t'} \log p(y_t|\theta_t) & \frac{\partial^2}{\partial \beta_t \partial \log \lambda_t} \log p(y_t|\theta_t) \\ \frac{\partial^2}{\partial \log \lambda_t \partial \beta_t'} \log p(y_t|\theta_t) & \frac{\partial^2}{\partial (\log \lambda_t)^2} \log p(y_t|\theta_t) \end{bmatrix}, \tag{25}$$

with entries

$$\begin{aligned} \frac{\partial^2}{\partial \beta_t \partial \beta_t'} \log p(y_t|\theta_t) &= -\Lambda(\lambda_t)' \Sigma_\epsilon^{-1} \Lambda(\lambda_t), \\ \frac{\partial^2}{\partial \log \lambda_t \partial \beta_t'} \log p(y_t|\theta_t) &= \big(y_t - \Lambda(\lambda_t)\beta_t\big)' \Sigma_\epsilon^{-1} \dot{\Lambda}(\lambda_t) - \big(\dot{\Lambda}(\lambda_t)\beta_t\big)' \Sigma_\epsilon^{-1} \Lambda(\lambda_t), \\ \frac{\partial^2}{\partial (\log \lambda_t)^2} \log p(y_t|\theta_t) &= \big(y_t - \Lambda(\lambda_t)\beta_t\big)' \Sigma_\epsilon^{-1} \ddot{\Lambda}(\lambda_t)\beta_t - \big(\dot{\Lambda}(\lambda_t)\beta_t\big)' \Sigma_\epsilon^{-1} \dot{\Lambda}(\lambda_t)\beta_t, \end{aligned}$$

where $\dot{\Lambda}(\lambda) = \frac{\partial \Lambda(\lambda)}{\partial \log \lambda}$ and $\ddot{\Lambda}(\lambda) = \frac{\partial^2 \Lambda(\lambda)}{\partial (\log \lambda)^2}$.

C Benchmark filters

C.1 Dynamic conditional score filter

The only difference between the ProPar filter and the dynamic conditional score (DCS) filter is in the update step. The update step of the DCS filter is analytically solvable, as the score is evaluated at the prediction instead of the update itself. Hence, the update step for the explicit filter is given by

$$\theta_{t|t} = \theta_{t|t-1} + H_t \nabla(y_t|\theta_{t|t-1}),$$

The prediction step that complements the update step is for the DCS filter the same as for the ProPar filter (given in Equation 18). For completeness, the prediction step is

$$\theta_{t+1|t} = \omega + \Phi\theta_{t|t}, \tag{26}$$

Both steps hold for the NS3F and NS4F model, but the factors included in θ_t and the dimensions of the static parameters are different.

C.2 Kalman filter

Although this thesis focuses on score-driven dynamic processes, the widely used Kalman filter operates on state-space models. This filter assumes that the time-varying factors are modelled via the following Gaussian autoregressive model:

$$\theta_{t+1} = \omega + \Phi\theta_t + \eta_{t+1}, \qquad \eta_{t+1} \sim \text{NID}(0, \Sigma_\eta),$$

where η_{t+1} is the $K \times 1$ vector of residuals with zero mean and variance matrix Σ_η.

As the Kalman filter is only applicable to models that possess linearity in the state vector, we distinguish the Kalman filter steps for the NS3F model (Appendix C.2.1) and the NS4F model (Appendix C.2.2). The latter model contains $\log \lambda_t$ as a fourth factor which introduces nonlinearities. The extended Kalman filter, which has already been applied to the NS4F model before by Koopman et al. (2010), locally linearizes the observation equation after which the regular Kalman filter can be applied.

C.2.1 Kalman filter in the NS3F model

In the linear three-factor Nelson-Siegel framework, the regular Kalman filter can be used. The Kalman update step for this model is given by

$$\begin{aligned}\beta_{t|t} &= \beta_{t|t-1} + B_{t|t-1}\Lambda'(\Lambda B_{t|t-1}\Lambda' + \Sigma_\epsilon)^{-1}(y_t - \Lambda\beta_{t|t-1}),\\ B_{t|t} &= B_{t|t-1} - B_{t|t-1}\Lambda'(\Lambda B_{t|t-1}\Lambda' + \Sigma_\epsilon)^{-1}\Lambda B_{t|t-1},\end{aligned}$$

where $\beta_{t|t}$ are the updated factors and $B_{t|t}$ the updated estimate of the mean squared error (MSE) matrix. The Kalman prediction step is denoted as

$$\begin{aligned}\beta_{t+1|t} &= \omega + \Phi\beta_{t|t},\\ B_{t+1|t} &= \Phi B_{t|t}\Phi' + \Sigma_\eta.\end{aligned}$$

Note that for the Kalman filter, we do not need to estimate any learning-rate matrix H_t. However, we do need an estimate for Σ_η. This estimate is obtained, together with the estimates for λ, Σ_ϵ, μ and Φ, by maximizing the prediction error decomposition.

C.2.2 Kalman filter in the NS4F model

Since allowing the shape parameter to be time-variant implies a nonlinear estimation system, we make use the extended Kalman filter. This extended filter locally linearizes the function $Z_t(\theta_t) = \Lambda(\lambda_t)\beta_t$ at $\theta_t = \theta_{t|t-1}$. The linearized model becomes

$$y_t = Z_t(\theta_{t|t-1}) + \breve{Z}_t \cdot (\theta_t - \theta_{t|t-1}) + \epsilon_t, \qquad \epsilon_t \sim \mathrm{NID}(0, \Sigma_\epsilon),$$

where $\breve{Z}_t = \frac{\partial Z_t(\theta)}{\partial\theta}\Big|_{\theta=\theta_{t|t-1}}$. The update steps are now defined as

$$\begin{aligned}\theta_{t|t} &= \theta_{t|t-1} + Q_{t|t-1}\breve{Z}_t'(\breve{Z}_t Q_{t|t-1}\breve{Z}_t' + \Sigma_\epsilon^+)^{-1}(y_t - Z(\theta_{t|t-1})),\\ Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1}\breve{Z}_t'(\breve{Z}_t Q_{t|t-1}\breve{Z}_t' + \Sigma_\epsilon^+)^{-1}\breve{Z}_t Q_{t|t-1},\end{aligned}$$

where $\theta_{t|t}$ are the updated factors and $Q_{t|t}$ the updated estimate of the mean squared error (MSE) matrix.

The prediction steps of the factors and the estimate of the MSE matrix are the same as in the linear NS3F model, and for completeness given by

$$\begin{aligned}\theta_{t+1|t} &= \omega + \Phi\theta_{t|t},\\ Q_{t+1|t} &= \Phi Q_{t|t}\Phi' + \Sigma_\eta.\end{aligned}$$

C.3 GAS-Kalman filter

An additional benchmark filter that we come up with, is a self-devised version of the Kalman filter. The filter is similar to the Kalman filter in the sense that it calculates the estimates of the states and their uncertainty matrices recursively over time using incoming measurements. However, the GAS-Kalman filter assumes the dynamics of the factors to be captured by the generalized autoregressive score (GAS) model instead of the vector autoregressive (VAR) model. The state-vector θ_t is modelled as

$$\theta_{t+1} = \omega + \Phi\theta_t + H_t \nabla(y_t|\theta_t).$$

Similar to the Kalman filter, we distinguish the prediction and update steps for the NS3F model (Appendix C.3.1) and the NS4F model (Appendix C.3.2), as the NS4F model introduces nonlinearities in the observation equation.

C.3.1 GAS-Kalman filter in the NS3F model

In the three-factor Nelson-Siegel framework, the update steps are the same as for the regular Kalman filter. For completeness, these steps are given by

$$\begin{aligned}\beta_{t|t} &= \beta_{t|t-1} + B_{t|t-1}\Lambda'(\Lambda B_{t|t-1}\Lambda' + \Sigma_\epsilon)^{-1}(y_t - \Lambda\beta_{t|t-1}),\\ B_{t|t} &= B_{t|t-1} - B_{t|t-1}\Lambda'(\Lambda B_{t|t-1}\Lambda' + \Sigma_\epsilon)^{-1}\Lambda B_{t|t-1},\end{aligned}$$

Prediction steps of the state-vector and its MSE matrix change to

$$\begin{aligned}\beta_{t+1|t} &= \mathbb{E}[\beta_{t+1}|\mathcal{F}_t]\\ &= \mathbb{E}\big[\omega + \Phi\beta_t + H_t\nabla(y_t|\beta_t)|\mathcal{F}_t\big]\\ &= \omega + \Phi\beta_{t|t} + H_t\nabla(y_t|\beta_{t|t})\\ B_{t+1|t} &= \mathbb{V}[\beta_{t+1}|\mathcal{F}_t]\\ &= \mathbb{V}\big[\omega + \Phi\beta_t + H_t\nabla(y_t|\beta_t)|\mathcal{F}_t\big]\\ &= \Phi B_{t|t}\Phi' + H_t\mathbb{V}\big[\Lambda'\Sigma_\epsilon^{-1}(y_t - \Lambda\beta_t)|\mathcal{F}_t\big]H_t'\\ &= \Phi B_{t|t}\Phi' + H_t\mathbb{V}\big[\Lambda'\Sigma_\epsilon^{-1}\Lambda\beta_t|\mathcal{F}_t\big]H_t'\\ &= \Phi B_{t|t}\Phi' + H_t\Lambda'\Sigma_\epsilon^{-1}\Lambda B_{t|t}\Lambda'\Sigma_\epsilon^{-1}\Lambda H_t',\end{aligned}$$

where $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$ denote the expectation and variance respectively, and $\mathcal{F}_t$ is the information set at time t. It is not common to use the score for prediction purposes. However, the assumed data generating process suggests that the one-step-ahead state depends on today's score. Therefore, the score evaluated at today's updated state contributes to the one-step-ahead prediction.

C.3.2 GAS-Kalman filter in the NS4F model

When we treat a fourth factor, $\log \lambda_t$, as time-varying in the Nelson-Siegel framework, the model becomes nonlinear in the state-vector. We locally linearize the function $Z(\theta_t) = \Lambda(\lambda_t)\beta_t$ at $\theta = \theta_{t|t-1}$. The update equations are the same as in the extended Kalman filter in Appendix C.2.2. For completeness, the update steps are given by

$$\begin{aligned} \theta_{t|t} &= \theta_{t|t-1} + Q_{t|t-1}\breve{Z}_t'(\breve{Z}_t Q_{t|t-1}\breve{Z}_t' + \Sigma_\epsilon^+)^{-1}(y_t - Z(\theta_{t|t-1})), \\ Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1}\breve{Z}_t'(\breve{Z}_t Q_{t|t-1}\breve{Z}_t' + \Sigma_\epsilon^+)^{-1}\breve{Z}_t Q_{t|t-1}, \end{aligned}$$

The prediction however changes as the state equation is different. The prediction equations of θ and its MSE estimate become

$$\begin{aligned} \theta_{t+1|t} &= \mathbb{E}[\theta_{t+1}|\mathcal{F}_t] \\ &= \mathbb{E}\big[\omega + \Phi\theta_t + H_t\nabla(y_t|\theta_t)|\mathcal{F}_t\big] \\ &= \omega + \Phi\theta_{t|t} + H_t\mathbb{E}\big[\nabla(y_t|\theta_t)|\mathcal{F}_t\big] \\ Q_{t+1|t} &= \mathbb{V}\big[\theta_{t+1}|\mathcal{F}_t\big] \\ &= \mathbb{V}\big[\omega + \Phi\theta_t + H_t\nabla(y_t|\theta_t)|\mathcal{F}_t\big] \\ &= \Phi Q_{t|t}\Phi' + H_t\mathbb{V}\big[\nabla(y_t|\theta_t)|\mathcal{F}_t\big]H_t'. \end{aligned}$$

Note that for the expressions $\mathbb{E}[\nabla(y_t|\theta_t)|\mathcal{F}_t]$ and $\mathbb{V}[\nabla(y_t|\theta_t)|\mathcal{F}_t]$ it is not straightforward to find an analytical solution as the elements of $\nabla(y_t|\theta_t)$ are nonlinear in θ_t. We therefore again locally linearize the function $\nabla(y_t|\theta_t)$ at $\theta_t = \theta_{t|t}$. We get

$$\nabla(y_t|\theta_t) = \nabla(y_t|\theta_{t|t}) + \nabla^2(y_t|\theta_{t|t})(\theta_t - \theta_{t|t}),$$

where $\nabla^2(y_t|\theta_{t|t}) = \frac{\partial\nabla(y_t|\theta_t)}{\partial\theta_t'}\Big|_{\theta_t=\theta_{t|t}}$ and hence

$$\begin{aligned} \mathbb{E}\big[\nabla(y_t|\theta_t)|\mathcal{F}_t\big] &= \mathbb{E}\big[\nabla(y_t|\theta_{t|t}) + \nabla^2(y_t|\theta_{t|t})(\theta_t - \theta_{t|t})|\mathcal{F}_t\big] = \nabla(y_t|\theta_{t|t}) \\ \mathbb{V}\big[\nabla(y_t|\theta_t)|\mathcal{F}_t\big] &= \mathbb{V}\big[\nabla(y_t|\theta_{t|t}) + \nabla^2(y_t|\theta_{t|t})(\theta_t - \theta_{t|t})|\mathcal{F}_t\big] = \nabla^2(y_t|\theta_{t|t})Q_{t|t}\nabla^2(y_t|\theta_{t|t})' \end{aligned}$$

resulting in the prediction steps

$$\begin{aligned}\theta_{t+1|t} &= \omega + \Phi\theta_{t|t} + H_t\nabla(y_t|\theta_{t|t}),\\ Q_{t+1|t} &= \Phi Q_{t|t}\Phi' + H_t\nabla^2(y_t|\theta_{t|t})Q_{t|t}\nabla^2(y_t|\theta_{t|t})'H_t'.\end{aligned}$$

D Simulation results benchmark models

Table 17: Out-of-sample yield forecasting performance

	VAR(1)	Naive forecast
NS3F model with GAS dynamics	0.219	0.309
NS3F model with state-space dynamics	0.757	0.798
NS4F model with GAS dynamics	1.449	0.316
NS4F model with state-space dynamics	2.388	0.552

Note: The table reports the out-of-sample yield forecasting performances of two benchmark models in the simulation study in terms of mean squared forecast error (MSFE). The benchmark models are the first-order vector autoregressive (*VAR[1]*) model and the naive forecast model. The models are applied to the three-and four-factor Nelson-Siegel framework in which the dynamics of the factors are generated using either a GAS model or a state-space model. Each of the 250 simulated data sets consist of $T = 1000$ observations. For the calculation of the MSFE, only the test samples are used, which contain the last 20% of each data set. The MSFE statistic that is displayed in the table is the average over four maturities, 200 test observations and 250 repetitions.