Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis MSc Quantitative Finance

Bellman Filtering for Dynamic Factor SV models

Thesis Proposal draft

Givani Boekestijn (582172)

Supervisor:	Rutger-Jan Lange
Second assessor:	Name of your second assessor
Date final version:	4th March 2025

Abstract

This thesis proposes a Dynamic Factor Stochastic Volatility (DFSV) framework for asset returns, incorporating VAR(1) dependencies in latent factor dynamics and volatilities, and a novel Bellman filtering approach for efficient hyperparameter estimation. I aim to extract latent factors driving asset returns while allowing their volatilities to evolve stochastically over time. The state-space model couples a vector autoregressive factor structure with time-varying volatility processes, capturing dynamic co-movement in both returns and risks. To estimate model parameters (hyperparameters) in this high-dimensional, non-linear setting, Bellman's dynamic programming principle is leveraged to perform approximate filtering and likelihood evaluation. Bellman filtering generalizes Kalman filtering to non-Gaussian, non-linear state-space models and yields a fast, stable estimator for latent states and an approximate likelihood for parameters (Lange (2024)). The expected contributions include theoretical extensions to dynamic factor volatility modeling and practical improvements in estimation speed and accuracy over traditional filters. I will empirically validate the model on financial datasets (e.g. Fama-French factor returns and sector indices), demonstrating its ability to capture time-varying covariances and improve risk forecasts.

Contents

Chapter 1

Introduction

Financial markets are characterized by time-varying returns and volatilities that often co-move across assets. Traditional asset pricing models like the Arbitrage Pricing Theory and Fama-French factor models assume static factor structures and often constant or independently modeled volatility. However, empirical evidence shows that both expected returns and volatilities vary over time and are partially predictable (Han, 2006). Capturing these dynamics is crucial for portfolio allocation, risk management, and asset pricing. Dynamic factor stochastic volatility models address this need by introducing latent factors that drive asset returns and allowing the volatility of these factors (and possibly idiosyncratic components) to change over time. This enriches factor models with a time-varying covariance structure, aligning with observed volatility clustering and factor correlations in markets.

Despite their appeal, dynamic factor models with stochastic volatility present significant estimation challenges. The state-space is high-dimensional when many assets and factors are involved, and the model is inherently non-linear and non-Gaussian due to the volatility dynamics. Classical estimation methods struggle in this context: Kalman filters require linear-Gaussian assumptions, and simulation-based methods (e.g. particle filters) face the curse of dimensionality, becoming computationally expensive or unstable as dimensionality grows (Lange, 2024). There is a clear motivation for new filtering and estimation techniques that can handle large-scale, non-linear state-space models efficiently.

This research is motivated by the gap between the sophistication of high-dimensional stochastic volatility models and the practicality of existing estimation methods. By exploring Bellman filtering – a recent innovation in state-space estimation – we aim to improve the extraction of dynamic factors and their volatilities from asset return data. The Bellman filter applies dynamic programming principles to state-space inference, potentially offering the accuracy of simulation-based filters at a fraction of the computational cost (Lange, 2024). Integrating this approach with a VAR(1)-driven factor volatility model is expected to yield a powerful framework for dynamic asset pricing, capturing both return co-movements and heteroskedastic risk in a coherent way. The following sections outline the relevant literature, our proposed methodology, and the plan for carrying out and validating this research.

Chapter 2

Literature Review

This chapter provides a comprehensive review of the existing literature related to the study. It begins with an examination of dynamic factor models as they pertain to financial returns, exploring how these models are formulated and applied in financial research. The chapter aims to contextualize the current study within established theoretical and empirical frameworks, identifying gaps and opportunities for further investigation.

2.1 Dynamic Factor Models for Financial Returns

Latent factor models have a long history in finance for explaining asset returns. Early examples include Ross's Arbitrage Pricing Theory and the Fama-French multi-factor models, which use static factors to summarize cross-sectional return drivers. Over time, researchers extended factor models to account for dynamics in the factors and their impact on returns. For example, Borghi, Hillebrand, Mikkelsen and Urga (2018) develop a two-level factor model with time-varying factor loadings and apply it to 1,815 global equities. They find that exposure ("beta") to global risk factors increases during worldwide shocks and that the persistence of these loadings depends on firm characteristics (e.g. larger firms exhibit less persistent factor exposures). Such results underscore that static factor models (like the Fama-French framework) may miss important temporal variation in risk exposures.

Aguilar and West (2000) introduced a Bayesian dynamic factor model with stochastic volatility for portfolio allocation, demonstrating improved modeling of time-varying risks in exchange rates. Their work showed that incorporating stochastic volatility in factor structures can enhance short-term forecast accuracy and portfolio decisions. Building on such ideas, Han (2006) proposed a Dynamic Factor Multivariate Stochastic Volatility (DFMSV) model that allows both the expected returns and volatilities of a large number of assets to vary over time. In this model, a few latent factors capture the common movements in returns, and crucially, each factor is assumed to follow an autoregressive process with time-varying volatility. This extension is motivated by empirical evidence that many economic predictors (e.g. dividend yields, interest rates) exhibit persistence, justifying AR(1) factor dynamics. Han (2006) finds that allowing factors to have stochastic volatilities and autocorrelations significantly improves the model's realism and its usefulness for asset allocation decisions.

Building on these foundations, more recent research has tackled the high-dimensional estimation challenges that come with dynamic factor SV models. Bayesian techniques are common: Mccausland (2015) and Kastner, Frühwirth-Schnatter and Lopes (2017), develop efficient Bayesian simulation methods that dramatically speed up estimation in factor SV models. By employing clever posterior simulation tricks (such as interweaving strategies to improve Markov chain Monte Carlo mixing), they report order-of-magnitude faster convergence without sacrificing accuracy. This makes it feasible to estimate factor volatility models on larger panels of assets (e.g. dozens of currencies or stocks) that were previously computationally prohibitive.

2.2 Stochastic Volatility in High Dimensions

Modeling time-varying volatility (heteroskedasticity) in multiple assets is challenging due to the curse of dimensionality. Traditional multivariate GARCH models quickly become overparameterized beyond a few series (Bollerslev, Engle & Nelson, 1994). A breakthrough came with Engle's Dynamic Conditional Correlation (DCC) model, which simplifies the covariance dynamics by modeling correlations separately from variances (Engle, 2000). The DCC model allows each asset's variance to follow a univariate GARCH process while the correlation matrix evolves in a parsimonious manner, making estimation feasible for dozens of series. This approach has been widely used in finance for its balance of flexibility and simplicity.

In stochastic volatility (SV) models provide an alternative by treating volatilities as latent variables following stochastic processes (often AR(1) in log-volatility). Multivariate SV models were explored by Harvey, Ruiz and Shephard (1994) and Jacquier, Polson and Rossi (1999) for small dimensions. To scale SV models to higher dimensions, researchers imposed factor structures. Philipov and Glickman (2006), for example, proposed a factor SV model where the covariance of factor returns follows a Wishart process. This Wishart multivariate SV framework allows a flexible yet structured evolution of the covariance matrix and demonstrated the ability to handle on the order of tens of assets by reducing effective dimensionality. Similarly, Yu and Meyer (2006) compared various multivariate SV specifications, finding that factor SV models often perform well in capturing co-movements. Asai, McAleer and Yu (2006) provide a comprehensive review of multivariate SV models, categorizing them into factor models, models with time-varying correlations, and others.

The factor-based SV models seem like a promising approach for modeling high-dimensional volatility. By assuming a few latent drivers for volatility (e.g., a market volatility factor, sector volatility factors, etc.), these models drastically reduce the number of parameters and mitigate estimation noise. The DFMSV model of Han (2006) is a prime example: it showed that with only a handful of latent factors, one can model the covariances of dozens of assets, something infeasible with unrestricted multivariate SV or GARCH. However, the complexity of such models (latent factors + latent time-varying variances) makes estimation a bottleneck. Early applications resorted to Bayesian MCMC estimation (which is simulation-intensive) or variational approximations. There remains a need for more efficient filtering and parameter estimation techniques to unlock the potential of high-dimensional SV models in practice.

2.3 Hyperparameter Estimation in State-Space Filtering

Estimating hyperparameters (such as the well known $ABCD$ matrices in the standard state-space formulation) is a critical step in implementing state-space models. In linear Gaussian state-space models, the classical approach is to use maximum likelihood via the Kalman filter and EM algorithm. Shumway and Stoffer (1982) pioneered the use of the EM algorithm with Kalman smoothing to iteratively find ML estimates of parameters (Shumway & Stoffer, 1982). Their method exploits the tractability of the linear Gaussian case, where the E-step computes expected sufficient statistics using Kalman smoother outputs, and the M-step updates parameter estimates analytically. This approach remains a workhorse for, e.g., linear dynamic factor models without stochastic volatility.

For non-linear or non-Gaussian models (such as those with stochastic volatility), parameter estimation is more challenging. One strategy is Bayesian estimation, treating hyperparameters as random and using MCMC or particle MCMC to sample from the joint posterior. Kim, Shephard and Chib (1998) and others applied Bayesian methods to univariate SV models successfully. However, fully Bayesian approaches can be very slow for high-dimensional problems.

Another strategy is to approximate the likelihood and use direct optimization (a quasi-ML approach). Extended Kalman Filters (EKF) and Unscented Kalman Filters (UKF) can provide on-line estimates of parameters by augmenting the state vector or by linearizing the state-space model, then applying ML or least squares on the residuals. These methods are faster but introduce bias if the linearization is crude. Moreover, EKF/UKF still assume certain model structures (e.g., additive Gaussian noise) and can perform poorly when those assumptions are violated (Lange, 2024).

Particle filtering (PF) offers a general solution by simulating many particles (state trajectories) and updating weights to approximate the filtering distribution. Particle filters can also estimate static parameters either by treating them as part of the state or via separate algorithms (like particle learning or particle EM). While very flexible, particle methods are notoriously resource-intensive for parameter estimation. The literature includes techniques like particle Markov-chain Monte Carlo and particle smoothing for likelihood evaluation. Malik and Pitt (2011) developed a particle filter approach for parameter estimation in more complex, non-gaussian SV models, but their method is practical only for low-dimensional cases (e.g. a single volatility process). In general, as the state dimension grows, particle filters suffer from weight degeneracy and require exponentially many particles to maintain accuracy.

Recent research has explored more efficient filtering approaches. Numerically accelerated importance sampling (NAIS) methods (e.g. (Koopman, Lucas & Scharth, 2015)) use analytical approximations to guide the particle filter, improving efficiency in moderate dimensions.

Approximations of the likelihood via Laplace methods or variational Bayes have also been considered for SV models. However, these either compromise some accuracy or still struggle when the model has many latent variables.

In summary, conventional hyperparameter estimation techniques either scale poorly with dimension (MCMC, particle filters) or rely on simplifying assumptions (EKF/UKF, analytic approximations). This motivates exploring the Bellman filtering approach, which constructs a pseudo-likelihood during filtering that can be optimized for parameter estimates ((Lange, 2024)). By circumventing heavy sampling and focusing on modes of the filtering distribution, Bellman filtering promises a faster route to hyperparameter identification, which is particularly appealing for high-dimensional, complex models.

2.4 Bellman Filtering vs. Traditional Methods

Bellman filtering is a recently developed method inspired by Bellman's dynamic programming principle for optimal control. In the context of state-space models, it recasts the filtering update as a sequential optimization problem rather than integration. At each time step, Bellman filtering seeks the mode of the posterior distribution of the state (or a quadratic approximation of the value function) by combining the prior (prediction from the previous state) with the new observation likelihood ((Lange, 2024)). This approach can handle non-linear and non-Gaussian features by effectively performing a proximal optimization step at each update. Lange (2024) shows that the Bellman filter generalizes the iterated Extended Kalman Filter to allow arbitrary observation and transition densities, while maintaining polynomial computational complexity. Notably, it avoids particle simulations, instead using a quadratic approximation to the log-posterior (similar in spirit to a Laplace approximation) at each time step.

The performance advantages reported for Bellman filtering are significant. First, it is computationally efficient, scaling on the order of $O(m^3)$ for state dimension m, similar to the Kalman filter. This makes it feasible for state vectors of size up to hundreds, which is often the case in factor models (e.g., a model with 10 factors and 100 idiosyncratic volatilities has a state dimension $m \approx 110$). In Lange's empirical studies, the Bellman filter was successfully applied to problems with $m \approx 150$. Second, it remains accurate despite the approximations: in simulation experiments for non-linear models, Bellman filtering achieved state estimation accuracy on par with particle filtering and other simulation-based methods, yet at a small fraction of the computational cost. In other words, it offers an excellent speed-accuracy trade-off, essentially bringing high-dimensional Bayesian filtering into a more practical regime.

Another advantage is stability. The Bellman filter update is shown to be a contraction in mean-square error under certain conditions, preventing the filter from diverging as time progresses. It also exhibits an invertibility property: the influence of the initial state guess decays exponentially fast. These properties are crucial when dealing with long financial time series, ensuring that estimation errors do not blow up over time – a common concern with naive implementations of EKF or particle filters.

Compared to the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), Bellman filtering does not assume Gaussian noise or rely on local linearization, which makes it more robust in cases of model misspecification or observation outliers. Unlike particle filters, it deterministically approximates the posterior, avoiding random Monte Carlo error and the need for a very large number of particles in high dimensions. Additionally, Bellman filtering naturally yields a pseudo-likelihood as a byproduct of its recursive updates. This pseudo-likelihood is exact for linear-Gaussian models and a second-order accurate approximation for general models. It can be maximized with respect to static parameters using standard optimization algorithms, thereby tackling the hyperparameter estimation problem in a quasi-maximum likelihood framework. This is a significant improvement over particle filters, which often require custom likelihood estimators or Bayesian approaches for parameter learning.

Chapter 3

Methodology

In this chapter, I describe the proposed framework for modeling asset returns using a dynamic factor stochastic volatility model. The model is designed to capture the common movements in asset returns through a small number of latent factors while accommodating time-varying volatility dynamics and interdependencies among these factors.

To address the high-dimensional and non-Gaussian nature of financial return data, I employ a state-space formulation and introduce the filters that will be applied to the model. In particular, the latent factors and their log-volatilities are both assumed to evolve according to a VAR(1) process. In addition, idiosyncratic risks are incorporated through asset-specific volatility terms. The noise will initially be modeled as Gaussian, but it is possible to modify the used distributions for the noise, like to a student-t distribution to model the heavier tails in returns. Finally, the methods for hyperparameter estimation will be discussed. In this section, I will specify the model and then outline the filtering and identification procedures.

3.1 Model Specification

3.1.1 Observation Equation and Factor Structure

Assume that the N-dimensional vector of asset returns, $\boldsymbol{r}_t$, is generated by a linear factor model with idiosyncratic SV returns:

$$\boldsymbol{r}_t = \Lambda \boldsymbol{f}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}\big(\mathbf{0},\ \Sigma_\epsilon\big), \tag{3.1}$$

$$\Sigma_\epsilon = \operatorname{diag}(\sigma_{1,t}, \dots, \sigma_{N,t}), \tag{3.2}$$

$$\sigma_{i,t} = e^{h_{i,t}^{(\mathrm{id})}}, \quad 1 \le i \le N \tag{3.3}$$

where:

- Λ is an $N \times K$ matrix of factor loadings.
- $\boldsymbol{f}_t = (f_{1,t}, \dots, f_{K,t})' \in \mathbb{R}^K$ is the vector of latent factors, with $K \ll N$, that captures the common variation among asset returns.

- $\boldsymbol{\epsilon}_t$ represents idiosyncratic errors with a diagonal covariance matrix, where the variance for asset i at time t is given by

$$\mathrm{Var}(\epsilon_{i,t}) = \sigma_{i,t}^2 = \exp\bigl(h_{i,t}^{(\mathrm{id})}\bigr), \quad i = 1, \dots, N. \tag{3.4}$$

This decomposition of asset returns into a common (factor) component and an idiosyncratic component is standard in asset pricing (see, e.g., (Chib, Nardari & Shephard, 2006)), and it allows for a significant dimensionality reduction.

3.1.2 State Equations: Factor Evolution and Log-Volatility Dynamics

To capture the time-varying risk and the persistence in the common factors, I model both the latent factors and their volatilities as evolving over time. In this framework, the asset returns are driven by a small number K of latent factors, and both the factors themselves and their associated volatilities exhibit serial correlation.

Factor Evolution

I assume that the latent factor vector $\boldsymbol{f}_t = (f_{1,t}, \dots, f_{K,t})^\top$ follows a VAR(1) process:

$$\boldsymbol{f}_{t+1} = \Phi_f\, \boldsymbol{f}_t + \boldsymbol{\nu}_{t+1}, \quad \boldsymbol{\nu}_{t+1} \sim \mathcal{N}\Bigl(\mathbf{0},\ \mathrm{diag}\Bigl(e^{h_{1,t+1}}, \dots, e^{h_{K,t+1}}\Bigr)\Bigr), \tag{3.5}$$

where Φ_f is a $K \times K$ matrix capturing the persistence and potential cross-factor influences. The innovation covariance is time-varying that depends on the latent log-volatilities $h_{k,t+1}$ for $k = 1, \dots, K$. Modeling the factors with an VAR(1) process allows us to capture persistence in the factor levels, which some studies (e.g., (Bernanke, Boivin & Eliasz, 2005)) have found to be significant, although many asset pricing models assume that the primary source of persistence comes from volatility clustering. Cross-factor dynamics are also possible here, because Φ_f can also be a full matrix.

Log-Volatility Dynamics

The latent log-volatilities of the factors, denoted by $\boldsymbol{h}_t = (h_{1,t}, \dots, h_{K,t})^\top$, are assumed to evolve jointly according to a VAR(1) process:

$$\boldsymbol{h}_{t+1} = \mu + \Phi_h\,(\boldsymbol{h}_t - \mu) + \boldsymbol{\eta}_{t+1}, \quad \boldsymbol{\eta}_{t+1} \sim \mathcal{N}(\mathbf{0},\, Q_h)\,, \tag{3.6}$$

where:

- $\boldsymbol{\mu} \in \mathbb{R}^K$ is the long-run mean vector of the log-volatilities (typically nonzero to reflect the empirical mean level of volatility),
- $\Phi_h da \in \mathbb{R}^{K\times K}$ is an autoregressive coefficient matrix that captures both the persistence in each factor's volatility and cross-factor spillovers,
- Q_h is a $K \times K$ positive-definite covariance matrix governing the volatility shocks.

This VAR(1) specification is more general than modeling each factor's volatility independently; it allows shocks to one factor's volatility to influence the volatilities of other factors, an interdependency that is empirically relevant in financial markets (see, e.g., (Han, 2006)).

The full state vector for factors and volatilities can then be written as:

$$x_t = \begin{pmatrix} \boldsymbol{f}_t \\ \boldsymbol{h}_t \end{pmatrix} \in \mathbb{R}^{2K}.$$

3.1.3 Incorporating Idiosyncratic Volatilities

While the systematic component of return volatility is captured by the dynamic factor structure, each asset also possesses an idiosyncratic risk component. As described in Eq. (3.4), the variance of the idiosyncratic error for asset i is modeled as:

$$\sigma_{i,t}^2 = \exp\big(h_{i,t}^{(\mathrm{id})}\big),$$

where $h_{i,t}^{(\mathrm{id})}$ is the idiosyncratic log-volatility. In the baseline specification, these idiosyncratic volatilities are assumed to be fixed over time to reduce complexity. This assumption is standard in many factor models (see, e.g., (Chib et al., 2006)), as the primary source of time variation is usually attributed to the common factors. Nonetheless, the framework is flexible enough to allow for a dynamic process (e.g., an AR(1) process) for $h_{i,t}^{(\mathrm{id})}$ if empirical evidence suggests time-varying idiosyncratic risk.

The overall covariance matrix of asset returns is then given by:

$$\Sigma_t = \Lambda \ \mathrm{diag}(e^{h_{1,t}}, \ldots, e^{h_{K,t}}) \, \Lambda' + \mathrm{diag}(e^{h_{1,t}^{(id)}}, \ldots, e^{h_{N,t}^{(id)}}). \tag{3.7}$$

This formulation distinguishes between systematic risk, driven by common factors, and asset-specific risk, captured by idiosyncratic volatilities. In high-dimensional settings, this decomposition is particularly valuable, as it reduces the effective number of parameters that need to be estimated while still capturing the key features of the return distribution.

This concludes the formulation, leading to a Dynamic Factor Stochastic Volatility (DFSV) model. Next, I will describe the filtering and estimation approach, including the implementation of Bellman filtering and comparisons with traditional filtering methods.

3.2 Filtering and Estimation Approach

In this section, I will describe the methodology for filtering the latent states and estimating the hyperparameters of the DFSV model. The main approach will be implementing a Bellman filter to jointly estimate the latent factors and their time-varying volatilities, while also extracting hyperparameters from the pseudo-likelihood produced during filtering. I will then compare this method to traditional filtering techniques, comparing its performance in high-dimensional, non-Gaussian settings.

3.2.1 Bellman Filtering for Joint State and Hyperparameter Estimation

Estimating the latent state $\boldsymbol{\alpha}_t$ at each time t given observations $\boldsymbol{y}_{s}{}_{s=1}^{t}$ is crucial for both inference and parameter learning. I will implement the Bellman filter as outlined by Lange (2024) to recursively approximate the filtering distribution $p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t})$. In each time step, Bellman filtering performs two main operations: Estimating the latent state

$$\boldsymbol{\alpha}_t = \begin{pmatrix} \boldsymbol{f}_t \\ \boldsymbol{h}_t \end{pmatrix} \in \mathbb{R}^{2K},$$

given the observed asset returns $\{\boldsymbol{r}_s\}_{s=1}^{t}$ is crucial for both inference and hyperparameter learning. I implement the Bellman filter, as outlined by (Lange, 2024), to recursively approximate the filtering distribution $p(\boldsymbol{\alpha}_t \mid \boldsymbol{r}_{1:t})$. At each time step, the Bellman filter performs two main operations:

1. Prediction (Time Update): I use the state-transition equations to predict a prior for $\boldsymbol{\alpha}_t$ based on the previous state estimate $\hat{\boldsymbol{\alpha}}_{t-1|t-1}$. The prediction step is given by:

$$\hat{\boldsymbol{\alpha}}_{t|t-1} = T\,\hat{\boldsymbol{\alpha}}_{t-1|t-1}, \tag{3.8}$$

$$\hat{P}_{t|t-1} = T\,\hat{P}_{t-1|t-1}\,T' + Q_t, \tag{3.9}$$

where:

- $T = \begin{bmatrix} \Phi_f & 0 \\ 0 & \Phi_h \end{bmatrix}$ is the block-diagonal state transition matrix, with Φ_f and Φ_h respectively governing the dynamics of the factors and log-volatilities.
- $Q_t = \begin{bmatrix} \mathrm{diag}\left(e^{h_{1,t}}, \dots, e^{h_{K,t}}\right) & 0 \\ 0 & Q_h \end{bmatrix}$ is the block-diagonal covariance matrix of the state innovations. Here, the factor innovations are scaled by the current factor volatilities $e^{h_{k,t}}$, while Q_h governs the shocks to the log-volatilities.

This prediction yields the prior state estimate $\hat{\boldsymbol{\alpha}}_{t|t-1}$ and its associated uncertainty $P_{t|t-1}$ before the new observation is taken into account.

2. Filtering (Measurement Update): When a new observation $\boldsymbol{r}_t$ becomes available, I incorporate it by augmenting the predicted value function with the log-likelihood of $\boldsymbol{r}_t$ given the state. For each time step, the following optimization is solved:

$$\hat{\boldsymbol{\alpha}}_{t|t} = \arg\max_{\boldsymbol{\alpha}_t \in \mathbb{R}^{2K}} \left\{ \ln p(\boldsymbol{r}_t \mid \boldsymbol{\alpha}_t) - \frac{1}{2}\left(\boldsymbol{\alpha}_t - \boldsymbol{a}_{t|t-1}\right)' \boldsymbol{I}_{t|t-1}^{-1} \left(\boldsymbol{\alpha}_t - \boldsymbol{a}_{t|t-1}\right) \right\}. \tag{3.10}$$

In this formulation, $\boldsymbol{\alpha}_t$ includes both the latent factors $\boldsymbol{f}_t$ and their log-volatilities $\boldsymbol{h}_t$. The observation likelihood $\ln p(\boldsymbol{r}_t \mid \boldsymbol{\alpha}_t)$ is computed from the model given by Eq. (3.1) and the conditional distribution of $\boldsymbol{f}_t$ from Eq. (3.5). The covariance matrix $\hat{P}_{t|t}$ is updated using the Fisher information matrix.

Hyperparameter Identification via Pseudo-Likelihood: I will also make use of the Bellman filter output for joint estimation of static hyperparameters Θ (which include the factor loadings Λ, the VAR matrices Φ_f and Φ_h, the long-run means $\boldsymbol{\mu}$, and the volatility innovation covariance Q_h). Lange (2024) provides an approximate pseudo-likelihood which can be used for parameter estimation based on the output of the Bellman filter as

$$\mathcal{L}(\Theta) \approx \sum_{t=1}^{T} \left\{ \ln p(\boldsymbol{r}_t \mid \hat{\boldsymbol{\alpha}}_{t|t}, \Theta) - \frac{1}{2} \log \frac{\det(P_{t|t-1})}{\det(P_{t|t})} - \frac{1}{2} \left(\hat{\boldsymbol{\alpha}}_{t|t} - \hat{\boldsymbol{\alpha}}_{t|t-1} \right)' P_{t|t-1}^{-1} \left(\hat{\boldsymbol{\alpha}}_{t|t} - \hat{\boldsymbol{\alpha}}_{t|t-1} \right) \right\}, \tag{3.11}$$

which is exact for linear Gaussian models and serves as a second-order approximation otherwise. I maximize $\mathcal{L}(\Theta)$ with respect to Θ using gradient-based optimization (e.g., BFGS). The gradient will be calculated using Automatic Differentiation to make calculations more computationally efficient. This process can also be embedded in an iterative Expectation-Maximization (EM) loop, where the Bellman filter provides the E-step (filtered/smoothed state estimates) and the M-step updates the hyperparameters.

3.2.2 Traditional Filtering methods

Traditional filtering approaches for state-space models include the Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filters (PF). The KF is optimal for linear Gaussian models but fails when the system is non-linear or non-Gaussian, as is the case here. The EKF and UKF address nonlinearity via linearization or sigma-point propagation, yet they can suffer from bias or instability in high-dimensional settings. While particle filters are capable of handling complex models, they often require a large number of particles to avoid degeneracy, resulting in high computational cost, particularly in high-dimensional systems (Kantas, Doucet, Singh, Maciejowski & Chopin, 2015).

In contrast, Bellman filtering reformulates the filtering problem as a sequence of optimization tasks, maintaining a computational complexity of $O(m^3)$ per time step (with m the state dimension), similar to the KF. This efficiency, along with its capacity to yield a pseudo-likelihood for hyperparameter estimation, makes the Bellman filter a robust and scalable alternative in high-dimensional, non-Gaussian contexts.

I anticipate that Bellman filtering will provide superior state estimation and parameter identification, both in terms of computational speed and accuracy, compared to traditional methods. This is particularly important for models like the DSFV model, where the state vector is high-dimensional and the return distribution can be modeled to exhibit pronounced non-Gaussian features.

3.3 Theoretical Contributions

In this research, I expect to make several contributions to the literature on financial econometrics and time series modeling.

My initial goal is to implement Bellman filtering—an approach rooted in dynamic programming—in the realm of high-dimensional financial volatility models. As far as I know, Bellman filtering has not been previously used for hyperparameter estimation in such intricate state-space models. This method will serve as a new template that others can use and modify for similar tasks.

I expect that utilizing a Bellman filter-based approach will considerably enhance computational efficiency and scalability in comparison to conventional methods such as particle filtering or Markov Chain Monte Carlo (MCMC). My objective is to prove that this method can efficiently manage extensive asset cross-sections and various factors on typical computing setups, thereby making advanced volatility models more attainable and applicable for practical implementations.

Through extensive simulation studies and empirical tests, I will compare the performance of my method with established filtering techniques. I expect to show that Bellman filtering not only produces more accurate parameter estimates and smoother volatility trajectories than the extended Kalman filter, but also operates orders of magnitude faster than particle filters. These concrete benchmarks will contribute to the ongoing methodological discourse in the field.

By applying my model to actual financial data—such as Fama-French factor returns and sector-specific indices—I aim to extract latent factors and their volatilities that reveal meaningful patterns, such as a latent "market volatility" factor or distinct sector cycles. These outputs can enhance risk forecasts, improve measures like Value-at-Risk or portfolio variance estimates, and ultimately inform more adaptive asset allocation strategies.

Chapter 4

Data

In this chapter I describe the data that I will use to test the DFSV model. I start by listing the data features needed to show the complex moreover changing nature of asset returns. Next I list the main data sources - such as Fama–French factor returns and sector or industry returns - that form the basis for both measuring the model besides judging its performance. Lastly I state the steps I follow to collect data, clean it along with carry out a basic review to confirm that the datasets remain neat, uniform along with fit for detailed econometric study.

4.1 Data Requirements

To effectively test and validate my model, the dataset must meet several important criteria. First, it should span a sufficiently long time horizon to capture a range of market regimes, from periods of relative calm to episodes of heightened volatility. This scope is critical for reliably estimating the dynamic behavior of both common factors and their stochastic volatilities. Second, the data must be high-dimensional, encompassing a broad cross-section of assets to demonstrate how the proposed filtering and estimation techniques perform at scale. Third, the dataset should include both systematic risk factors—such as Fama–French factor returns—and granular asset returns that can be organized by sector or industry. Such a structure is essential for separating returns into common and idiosyncratic components. Finally, the dataset must be of high quality, with minimal missing values and anomalies, so that the state-space model and subsequent filtering procedures are not unduly compromised by data irregularities.

4.2 Data Sources

For my empirical analysis, I will rely on two main data sources:

Fama–French Factor Returns: The Fama-French Data library (*Kenneth R. French - Data Library*, n.d.) contains both portfolios sorted by factors such as Size and Book-to-Market ratio and portfolios sorted by industry. It is well established in the asset pricing literature (see, e.g., (Han, 2006)) and will be used both as input for dynamic factor extraction and as a benchmark for evaluating model performance.

Sector/Industry Returns: Apart from the Fama-French factors, sector or industry portfolio returns (or individual stocks allocated to each sector) will be collected. This approach will help me check whether the model is capable of capturing other, more granular, asset-specific risks that typical factors do not consider. Reliable financial databases like Bloomberg, CRSP, and Yahoo Finance can be sourced for sector return data. This data is very important in validating the decomposition of asset returns into their systematic and idiosyncratic components and assessing the model for its predictive performance in real-world applications.

4.3 Data Collection, Processing, and Preliminary Analysis

The data collection process involves several steps to ensure that the datasets are robust and suitable for analysis. First, I will acquire the Fama–French factor returns from the Kenneth French Data Library, which provides monthly (and sometimes daily) data over several decades. Similarly, I will obtain sector or industry return data from a reliable financial database.

Once the raw data is collected, I will perform extensive cleaning and preprocessing. This includes:

- **Alignment:** Synchronize all time series to a common frequency (e.g., monthly or weekly) and address missing values by means of interpolation or exclusion.
- **Normalization:** Adjust for corporate actions, dividends, and other market events, ensuring that return series remain consistent across time and assets.
- **Outlier Detection:** Identify and manage extreme observations that could otherwise distort volatility estimates.
- **Preliminary Statistical Analysis:** Perform exploratory data analysis (EDA) to compute summary statistics, visualize time series behavior, and gauge initial correlations. I will also use principal component analysis (PCA) to estimate how many latent factors the model should include.

These preprocessing steps are essential to ensure that the state-space model is estimated using high-quality data. In addition to validating data integrity, the preliminary analysis uncovers the dynamic behavior of returns and volatilities, which will guide both the model's specification and the choice of filtering methods.

By undertaking these measures, I aim to create a robust, high-dimensional dataset that spans various market regimes. In doing so, I establish a solid empirical foundation for evaluating the performance of my proposed model.

Chapter 5

Empirical Validation Plan

5.1 Simulation Study Design

In order to test the DFSV model under controlled conditions, I will run a detailed simulation study. By creating synthetic data based on the same setup described in my Methodology section, I can directly observe the "true" parameters and states. This approach allows me to measure how accurately the model and its filtering process can estimate those known values. Below is a summary of the main experiments I plan to conduct:

Scaling with Number of Assets(N) and Factors (K) : I will simulate asset return data for various model sizes, (from small scale with $N = 10$ assets and $K = 1$–2 factors, up to larger scales like $N = 100+$ and $K = 5$) to examine how the Bellman filter performs compared to tradional filtering methods as dimensionality grows.

Performance metrics will include estimation accuracy of the latent states and filter stability. I will track how estimation error and convergence behaviour change with increasing N and K, as well as the computational time required. This will demonstrate the model's scalability and highlight any practical limits on the number of assets or factors.

Baseline vs. Heavy-Tailed Innovations: Under the baseline model, simulation errors will follow a normal distribution (consistent with the model's assumptions). In another experiment, the model can be adapted to introduce Student-t distributed innovations (with degrees of freedom chosen to induce heavy tails) to assess robustness to non-normal return behaviour.

By comparing results from Gaussian and Student-t simulations, I can determine if the Bellman filter remains effective when returns exhibit fat tails. If performance degrades significantly under Student-t noise, this would motivate potential model extensions like incorporating a heavy-tail noise assumption or adjustments in the filtering approach.

Varying Factor Loadings Structures: I will also investigate how different factor loading patterns influence estimation quality. Some scenarios will feature diversified loadings (where each asset has moderate exposure to multiple factors), while others will include concentrated loadings (where each factor strongly affects only a subset of assets). Additionally, I will simulate cases in which factors are highly correlated versus nearly orthogonal. This variety will help me see how well the Bellman filter can identify factor structures and track volatility dynamics, even under diverse loading setups.

5.1.1 Evaluation in the Simulation Study:

Within each experiment, I will measure parameter estimation error—specifically, the difference between true and estimated factor loadings, as well as volatility dynamics coefficients. I will also gauge state estimation accuracy, such as how close the estimated volatilities are to their true simulated values. Beyond that, I will keep track of the log-likelihood or filtering error over time to see how quickly the filter adapts. These findings will clarify the strengths and weaknesses of the Dynamic Factor Stochastic Volatility (DFSV) model with Bellman filtering, and they will guide any adjustments needed before I apply the model to real-world data.

5.2 Application to Real-World Data

After confirming the model's performance through simulation, I will apply it to real financial data to see how it holds up in practice. As described in the Data section, these datasets include the Fama–French factor returns and a collection of sector or asset returns, and they serve as the main testing ground for out-of-sample predictions and portfolio performance.

Predictive Performance (MSPE): To gauge how well the DFSV model with a filter for latent state estimation is able to forecast future volatility and/or returns, one-step ahead predictions will be generated after fitting the model. With these, I can calculate the Mean Squared Prediction Error (MSPE) using the conditional distribution of asset returns and compare it to other models. I will use two simpler benchmarks: (1) a univariate GARCH(1,1) model applied independently to each asset, and (2) a static factor model that assumes constant volatility over time. A lower MSPE indicates stronger predictive accuracy. I expect the dynamic factor volatility framework to outperform a constant-volatility factor model by capturing time-varying risk, and to at least match or surpass GARCH by exploiting common factors across different assets. The performance of the Bellman filter compared to tradional filtering methods is also of interest here, both in accuracy and computational efficiency.

Portfolio Covariance and Sharpe Ratio Evaluation: Beyond predicting individual returns, another practical test is the quality of the estimated covariance matrices for portfolio construction. Using the model's filtered estimates of the time-varying covariance of asset returns, we will construct mean–variance efficient portfolios (for instance, the global minimum-variance portfolio or a tangency portfolio if we have an estimate of expected returns).

We will then measure the out-of-sample Sharpe ratios (risk-adjusted returns) of these portfolios over the test period. The same exercise will be done using covariance estimates from simpler models – for example, using a constant covariance matrix (such as the sample covariance or a constant factor covariance) and using univariate GARCH volatilities with an assumption of constant correlations. Comparing Sharpe ratios (and portfolio volatility/return characteristics) will show whether the dynamic factor volatility model provides tangible improvement in investment performance. A higher Sharpe ratio obtained would indicate that better covariance timing and risk capture translate into more efficient portfolios.

Throughout the real-data application, I will perform rolling or updating forecasts to mimic real-time use: re-estimating the model periodically as new data arrives and then forecasting the next period. This process aligns with how an investor or risk manager would use the model. All results (forecast errors and portfolio metrics) will be statistically analyzed to confirm if differences in performance are significant.

5.3 Benchmarking and Comparative Analysis

To put the Bellman filtering approach into perspective, I will benchmark it against traditional filtering and estimation techniques. These benchmarks will be done in parallel to the previous two sections. This comparative analysis will clarify the benefits and trade-offs of using Bellman filtering for the DFSV model. By organizing the comparison by method, efficiency, accuracy, and robustness, I can capture a full range of performance metrics.

Filtering Method Comparison: Alongside the Bellman filter, I will implement alternative state-space methods—such as the Extended Kalman Filter (EKF) and a Particle Filter (PF)—to handle the DFSV model. The EKF relies on linearizing nonlinear volatility dynamics, while the PF uses simulation-based Bayesian filtering. Applying these techniques to the same simulated and real datasets will help me compare their accuracy in tracking latent volatilities, as well as their stability in higher-dimensional settings. Metrics like filter log-likelihood and latent state estimation error will indicate whether Bellman filtering provides any advantage, and how each method handles nonlinearities.

Computational Efficiency: I will record the computational performance of each approach, focusing on total run time, per-time-step computation, and convergence speed during parameter estimation. Since high-dimensional filtering can be computationally demanding, I will observe how quickly each method scales as the number of assets (N) and factors (K) increases. This analysis will show whether Bellman filtering's potentially improved accuracy (if any) comes at a reasonable cost or whether it offers efficiency gains compared to the EKF and PF.

Estimation Accuracy and Convergence: Beyond filtering the latent states, the DFSV model also involves estimating static parameters, such as factor loadings and volatility dynamics. In simulation, I will measure how closely each method's estimates match the true values. In real data, I will compare log-likelihoods (or posterior likelihoods) to see if any method attains a better fit. Convergence behavior will be monitored for evidence of local optima or parameter instability. By evaluating how each technique balances accurate parameter estimates with a smooth optimization process, I can determine whether Bellman filtering's design yields an edge in practice.

Robustness Checks (Noise and Distributional Assumptions): Finally, I will test how each method performs under stressed conditions. This may involve injecting noise, outliers, or heavy-tailed distributions—either through extreme market shocks or by simulating fat-tailed returns. I will also check the impact of model misspecifications, such as an incorrect number of factors or slightly different dynamics than assumed. Examining whether the Bellman filter adapts more effectively than the EKF or PF in these scenarios will indicate each method's reliability when real-world data deviates from ideal assumptions.

After these comparisons, I hope to provide a clear and balanced view of whether Bellman filtering is superior, on par, or less effective than alternative approaches. The conclusions will tie back to earlier methodological choices, confirming (or challenging) the hypothesis that Bellman filtering can be a robust and efficient method for high-dimensional dynamic factor stochastic volatility models.

Chapter 6

Time Frame and Project Timeline

Table 6.1: Week-by-Week Thesis Roadmap

Week 1	**Problem Refinement, Model Structure, and Simulation Setup:** Refine the problem statement and finalize the dynamic factor SV model structure (e.g., initial factor count). Formally specify the state-space equations, and prepare a simple data-generating process (DGP) for initial testing. Develop preliminary code (in Python) for simulating the DGP, and outline the Bellman filter algorithm in pseudo-code.
Week 2	**Implementing the Bellman Filter Algorithm:** Construct the Bellman filter for a simple nonlinear state-space model, starting with a univariate SV example to validate basic functionality. Compare results against known methods (e.g., EKF), then extend to handle multiple factors or assets. Apply the filter to the small DGP from Week 1, addressing issues related to convergence or optimization. By the end of this week, ensure stable state estimates in simplified cases.
Week 3	**Hyperparameter Estimation Module:** Develop routines to compute the pseudo-likelihood from the Bellman filter output. Implement an optimization approach (e.g., gradient-based or `scipy` optimizers) to maximize this likelihood. Test on simulated data with known parameters to confirm estimation accuracy. Address any stability concerns (e.g., adding regularization or handling factor label swapping). By the end of this week, have an end-to-end pipeline—filtering plus parameter estimation—on simulated data.
Week 4	**Empirical Data Preparation:** Start documenting results of simulation studies. Obtain Fama–French factor data and sector (or asset) return data. Clean and align these datasets to consistent frequencies (e.g., monthly or weekly). Perform exploratory data analysis: compute summary statistics, correlation matrices, and possibly run a PCA to guide factor selection. Convert data into model-compatible formats, and if time allows, conduct a preliminary run of the Bellman filter on a small subset of real data to identify issues.
Week 5	**Model Application and Tuning on Real Data:** Run the Bellman filter and hyperparameter estimation on the Fama–French factor data. Monitor convergence, adjusting optimization settings if needed. Once stable, apply the same approach to the sector or asset returns. Optimize code performance for higher-dimensional data, and document outcomes (e.g., estimated factors, volatilities, log-likelihoods). Start documenting results of empirical studies.
Week 6	**Benchmarking & Analysis:** Compare the proposed model's performance to benchmarks like DCC-GARCH, EKF, or particle filters. Assess model fit (AIC/BIC), forecast accuracy (e.g., RMSE, VaR exceedance), and interpret the results. Refine the model or data handling if results are unexpected. Perform robustness checks (subperiod estimations, modifications of assumptions), and produce tables and figures. Document benchmark results.
Week 7	**Thesis Writing & Finalization:** Compile methodology, findings, and analyses into the thesis document. Integrate all relevant elements: problem context, model details, empirical outcomes, and comparisons. Prepare visuals (tables, plots) for clarity. Perform final proofreading and ensure references are complete. By the end of this week, finalize the thesis, presenting a coherent narrative from problem statement to conclusion.

References

Aguilar, O. & West, M. (2000). Bayesian Dynamic Factor Models and Portfolio Allocation. *Journal of Business & Economic Statistics*, *18*(3), 338–357. Retrieved 2025-03-04, from `https://www.jstor.org/stable/1392266` (Publisher: [American Statistical Association, Taylor & Francis, Ltd.]) doi: 10.2307/1392266

Asai, M., McAleer, M. & Yu, J. (2006, September). Multivariate Stochastic Volatility: A Review. *Econometric Reviews*, *25*(2-3), 145–175. Retrieved 2025-03-04, from `https://doi.org/10.1080/07474930600713564` (Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07474930600713564) doi: 10.1080/07474930600713564

Bernanke, B. S., Boivin, J. & Eliasz, P. (2005). Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach. *The Quarterly Journal of Economics*, *120*(1), 387–422. Retrieved 2025-03-02, from `https://www.jstor.org/stable/25098739` (Publisher: Oxford University Press)

Bollerslev, T., Engle, R. F. & Nelson, D. B. (1994, January). Chapter 49 Arch models. In *Handbook of Econometrics* (Vol. 4, pp. 2959–3038). Elsevier. Retrieved 2025-03-04, from `https://www.sciencedirect.com/science/article/pii/S1573441205800182` doi: 10.1016/S1573-4412(05)80018-2

Borghi, R., Hillebrand, E., Mikkelsen, J. & Urga, G. (2018, December). The dynamics of factor loadings in the cross-section of returns. *The dynamics of factor loadings in the cross-section of returns*. (Place: Aarhus Publisher: Institut for Økonomi, Aarhus Universitet)

Chib, S., Nardari, F. & Shephard, N. (2006, October). Analysis of high dimensional multivariate stochastic volatility models. *Journal of Econometrics*, *134*(2), 341–371. Retrieved 2025-03-02, from `https://linkinghub.elsevier.com/retrieve/pii/S0304407605001478` doi: 10.1016/j.jeconom.2005.06.026

Engle, R. F. (2000). Dynamic Conditional Correlation - A Simple Class of Multivariate GARCH Models. *SSRN Electronic Journal*. Retrieved 2025-03-04, from `http://www.ssrn.com/abstract=236998` doi: 10.2139/ssrn.236998

Han, Y. (2006). Asset Allocation with a High Dimensional Latent Factor Stochastic Volatility Model. *The Review of Financial Studies*, *19*(1), 237–271. Retrieved 2025-02-28, from `https://www.jstor.org/stable/3598036` (Publisher: [Oxford University Press, Society for Financial Studies])

Harvey, A., Ruiz, E. & Shephard, N. (1994, April). Multivariate Stochastic Variance Models. *The Review of Economic Studies*, *61*(2), 247–264. Retrieved 2025-03-04, from `https://doi.org/10.2307/2297980` doi: 10.2307/2297980

Jacquier, E., Polson, N. G. & Rossi, P. (1999, July). *Stochastic Volatility: Univariate and*

Multivariate Extensions (CIRANO Working Paper). CIRANO. Retrieved 2025-03-04, from https://econpapers.repec.org/paper/circirwor/99s-26.htm

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J. & Chopin, N. (2015, August). On Particle Methods for Parameter Estimation in State-Space Models. *Statistical Science*, *30*(3), 328–351. Retrieved 2025-03-03, from https://projecteuclid.org/journals/statistical-science/volume-30/issue-3/On-Particle-Methods-for-Parameter-Estimation-in-State-Space-Models/10.1214/14-STS511.full (Publisher: Institute of Mathematical Statistics) doi: 10.1214/14-STS511

Kastner, G., Frühwirth-Schnatter, S. & Lopes, H. F. (2017, October). Efficient Bayesian Inference for Multivariate Factor Stochastic Volatility Models. *Journal of Computational and Graphical Statistics*, *26*(4), 905–917. Retrieved 2025-03-04, from https://doi.org/10.1080/10618600.2017.1322091 (Publisher: ASA Website _eprint: https://doi.org/10.1080/10618600.2017.1322091) doi: 10.1080/10618600.2017.1322091

Kenneth R. French - Data Library. (n.d.). Retrieved 2025-03-03, from https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Kim, S., Shephard, N. & Chib, S. (1998). Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models. *The Review of Economic Studies*, *65*(3), 361–393. Retrieved 2025-03-04, from https://www.jstor.org/stable/2566931 (Publisher: [Oxford University Press, Review of Economic Studies, Ltd.])

Koopman, S. J., Lucas, A. & Scharth, M. (2015, January). Numerically Accelerated Importance Sampling for Nonlinear Non-Gaussian State-Space Models. *Journal of Business & Economic Statistics*, *33*(1), 114–127. Retrieved 2025-03-04, from https://doi.org/10.1080/07350015.2014.925807 (Publisher: ASA Website _eprint: https://doi.org/10.1080/07350015.2014.925807) doi: 10.1080/07350015.2014.925807

Lange, R.-J. (2024, January). Bellman filtering and smoothing for state–space models. *Journal of Econometrics*, *238*(2), 105632. Retrieved 2024-03-28, from https://www.sciencedirect.com/science/article/pii/S0304407623003482 doi: 10.1016/j.jeconom.2023.105632

Malik, S. & Pitt, M. K. (2011, February). *Modelling Stochastic Volatility with Leverage and Jumps: A Simulated Maximum Likelihood Approach via Particle Filtering* [SSRN Scholarly Paper]. Rochester, NY: Social Science Research Network. Retrieved 2025-03-04, from https://papers.ssrn.com/abstract=1763783 doi: 10.2139/ssrn.1763783

Mccausland, W. J. (2015). DYNAMIC FACTOR MODELS WITH STOCHASTIC VOLATILITY.

Philipov, A. & Glickman, M. E. (2006, September). Factor Multivariate Stochastic Volatility via Wishart Processes. *Econometric Reviews*, *25*(2-3), 311–334. Retrieved 2025-03-04, from https://doi.org/10.1080/07474930600713366 (Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07474930600713366) doi: 10.1080/07474930600713366

Shumway, R. H. & Stoffer, D. S. (1982). An Approach to Time Series Smoothing and Forecasting Using the Em Algorithm. *Journal of Time Series Analysis*, *3*(4), 253–264. Retrieved 2025-03-04, from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.1982.tb00349.x (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9892.1982.tb00349.x) doi: 10.1111/j.1467-9892.1982.tb00349.x

Yu, J. & Meyer, R. (2006, September). Multivariate Stochastic Volatility Models: Bayesian Estimation and Model Comparison. *Econometric Reviews*, *25*(2-3), 361–384. Retrieved 2025-03-04, from https://doi.org/10.1080/07474930600713465 (Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07474930600713465) doi: 10.1080/07474930600713465