

TI 2020-052/III  
Tinbergen Institute Discussion Paper

# Bellman filtering for state-space models

*Rutger-Jan Lange<sup>1</sup>*

<sup>1</sup> Econometric Institute, Erasmus School of Economics, Rotterdam, Netherlands

Tinbergen Institute is the graduate school and research institute in economics of Erasmus University Rotterdam, the University of Amsterdam and Vrije Universiteit Amsterdam.

Contact: [discussionpapers@tinbergen.nl](mailto:discussionpapers@tinbergen.nl)

More TI discussion papers can be downloaded at <https://www.tinbergen.nl>

Tinbergen Institute has two locations:

Tinbergen Institute Amsterdam  
Gustav Mahlerplein 117  
1082 MS Amsterdam  
The Netherlands  
Tel.: +31(0)20 598 4580

Tinbergen Institute Rotterdam  
Burg. Oudlaan 50  
3062 PA Rotterdam  
The Netherlands  
Tel.: +31(0)10 408 8900

# Bellman filtering for state-space models

Rutger-Jan Lange\*  
Econometric Institute, Erasmus School of Economics  
Rotterdam, Netherlands

August 27, 2020

## Abstract

This article presents a new filter for state-space models based on Bellman’s dynamic programming principle applied to the posterior mode. The proposed Bellman filter generalises the Kalman filter including its extended and iterated versions, while remaining equally inexpensive computationally. The Bellman filter is also (unlike the Kalman filter) robust under heavy-tailed observation noise and applicable to a wider range of models. Simulation studies reveal that the mean absolute error of the Bellman-filtered states using estimated parameters typically falls within a few percent of that produced by the mode estimator evaluated at the true parameters, which is optimal but generally infeasible.

JEL Classification Codes: C32, C53, C61

Keywords: Bellman filter, dynamic programming, Kalman filter, maximum a posteriori (MAP) estimate, posterior mode, state-space model

## 1 Introduction

State-space models allow observations to be affected by a hidden state that changes stochastically over time. For discrete times  $t = 1, 2, \dots, n$ , suppose the observation  $\mathbf{y}_t \in \mathbb{R}^l$  is drawn from a given observation density,  $p(\mathbf{y}_t|\boldsymbol{\alpha}_t)$ , while the latent state  $\boldsymbol{\alpha}_t \in \mathbb{R}^m$  follows a first-order Markov process with a known state-transition kernel,  $p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})$ , and some given initial condition,  $p(\boldsymbol{\alpha}_1)$ , i.e.

$$\mathbf{y}_t \sim p^\psi(\mathbf{y}_t|\boldsymbol{\alpha}_t), \quad \boldsymbol{\alpha}_t \sim p^\psi(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}), \quad \boldsymbol{\alpha}_1 \sim p^\psi(\boldsymbol{\alpha}_1). \quad (1)$$

Here  $p(\cdot|\cdot)$  and  $p(\cdot)$  denote generic conditional and marginal densities; i.e. any two  $p$ ’s need not denote the same probability density function (e.g. Polson et al., 2008, p. 415). The dependence on a fixed (hyper)parameter  $\psi$  is indicated by the superscript. Both the observation and state-transition densities may be non-Gaussian and involve nonlinearity. Sequences of observations are denoted  $\mathbf{y}_{1:t} := (\mathbf{y}_1, \dots, \mathbf{y}_t) \in \mathbb{R}^{l \times t}$ ; likewise for the states, i.e.  $\boldsymbol{\alpha}_{1:t} := (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_t) \in \mathbb{R}^{m \times t}$ .

---

\*I thank Maksim Anisimov, Francisco Blasques, Dick van Dijk, Dennis Fok, Simon Godsill, Maria Grith, Andrew Harvey, Siem Jan Koopman, André Lucas, Andrea Naghi, Jochem Oorschot, Richard Paap, Andreas Pick, Krzysztof Postek, Rogier Quadvlieg, Daniel Ralph, Annika Schnücker, Stephen Thiele, Michel van der Wel, Martina Zaharieva, Mikhail Zhelonkin and Chen Zhou for valuable feedback.

Myriad examples of model (1) can be found in signal processing, image analysis, speech recognition, DNA sequence analysis, oceanography, and time-series modelling (see Yun et al., 2017, and the references therein). In the field of financial econometrics, examples include dynamic models for count data (Singh and Roberts, 1992; Frühwirth-Schnatter and Wagner, 2006), intensity (Bauwens and Hautsch, 2006), duration (Bauwens and Veredas, 2004), volatility (Jacquier et al., 2002; Tauchen and Pitts, 1983; Harvey et al., 1994; Taylor, 2008; Ghysels et al., 1996) and dependence structure (Hafner and Manner, 2012).

Model (1) confronts researchers with two important problems, known as the filtering and estimation problems. The filtering problem treats the constant (hyper)parameter  $\boldsymbol{\psi}$  as known, while the latent states  $\boldsymbol{\alpha}_{1:t}$  are to be estimated, possibly in real time. The estimation problem entails determining the parameter  $\boldsymbol{\psi}$ , where both this parameter and the latent states  $\boldsymbol{\alpha}_{1:t}$  are assumed to be unknown. The filtering problem can be solved in closed form only when considering (a) linear Gaussian models (Kalman, 1960) or (b) Markov-switching models where the state can take a finite number of (discrete) values (Hamilton, 1989). Most real-world problems, however, fall outside these special classes. To address such filtering problems, popular approaches include ensemble methods (e.g. Chen and Liu, 2000; Stroud et al., 2003; Katzfuss et al., 2019) and simulation techniques (e.g. Fearnhead and Clifford, 2003; Godsill et al., 2004; De Valpine, 2004; Lin et al., 2005; Andrieu et al., 2010; Bunch and Godsill, 2016; Guarniero et al., 2017; Jacob et al., 2019).

This article introduces the Bellman filter, which, to achieve computational simplicity, is based not on simulation but on approximation. Specifically, the Bellman filter approximates, in real time, the *maximum a posteriori* (MAP, e.g. Godsill et al., 2001; Sardy and Tseng, 2004, p. 191) estimate of the most recent state. The MAP estimate is also known as the posterior mode (e.g. Fahrmeir and Kaufmann, 1991; Fahrmeir, 1992; Shephard and Pitt, 1997; Durbin and Koopman, 1997; So, 2003; Jungbacker and Koopman, 2007). The label ‘posterior’ does not reflect a Bayesian approach; it indicates only that the mode is computed after the data are received. The mode maximises the joint likelihood of the states and the data, denoted as  $p(\boldsymbol{\alpha}_{1:t}, \mathbf{y}_{1:t})$ , or, equivalently, the conditional density  $p(\boldsymbol{\alpha}_{1:t} | \mathbf{y}_{1:t})$ . Here, the data  $\mathbf{y}_{1:t}$  are considered known, while the states  $\boldsymbol{\alpha}_{1:t} \in \mathbb{R}^{m \times t}$  are variables to be optimised (when the states represent variables rather than realisations, we write them in Roman font to distinguish them from the true states  $\boldsymbol{\alpha}_{1:t}$ ). The mode is attractive because it “has an optimality property analogous to that of maximum likelihood estimates of fixed parameters in finite samples” (Durbin and Koopman, 2012, pp. 252-3). Further, the mode is the logical choice when considering zero-one loss functions, as in e.g. target-tracking applications (Godsill et al., 2001).

However, obtaining the mode in real time is computationally cumbersome, as it involves re-estimating the entire sequence of states  $\boldsymbol{\alpha}_{1:t}$  for each time  $t$ . Computing times typically increase exponentially as time progresses. Moreover, the mode fails to address the estimation problem, making its application infeasible in practice unless supplemented with other methods. These drawbacks may explain why, despite its optimality properties, the mode has to date received little attention as a filter.

In this article, we circumvent both drawbacks of the mode, yielding an algorithm that is both fast and feasible, while performing on par with the mode estimator in simulation studies. For simplicity, we assume the mode exists (and is unique), and we take its optimality properties for granted. To address the first drawback of the mode — computational complexity — we employ Bellman’s

principle of dynamic programming (Bellman, 1957). This allows us to generate a computable sequence of filtered states without having to solve optimisation problems involving an ever-increasing number of states. Instead, Bellman’s equation involves maximisation over a single state for each time step, vastly reducing the computational burden (from being exponentially increasing to being constant over time).

For linear Gaussian state-space models, Bellman’s equation can be solved exactly, yielding, perhaps unsurprisingly, Kalman’s (1960) filter. In this case, the solution to Bellman’s equation, known as the *value function*, turns out to be multivariate quadratic, with a unique global maximum for every time step. The argmax equals the Kalman-filtered state, while the inverse of the negative Hessian equals the Kalman-filtered covariance matrix. Although it cannot generally be solved in closed form, Bellman’s equation remains valid for the entire class of state-space models (1). Drawing on the field of approximate dynamic programming (ADP, e.g. Bertsekas, 2012) opens the door to the development of a new class of filtering techniques that are accurate and do not require simulation of any kind.

To illustrate the workings of the Bellman filter, we focus on state-space models where the state dynamics remain linear and Gaussian; this class is still general enough for many practical purposes. We use ADP methods to approximate the value function, for each time step, using a multivariate quadratic function. If the model were linear and Gaussian, this approximation would be exact. In general, a multivariate quadratic function can still approximate, at least locally, any value function that is smooth in the vicinity of its global maximum. As with the Kalman filter, the researcher keeps track of a filtered state and an associated precision matrix, which, in the case of the Bellman filter, are determined by the argmax of the value function and the negative Hessian matrix evaluated at the peak. The quadratic approximation is simple enough to yield a computationally efficient filter, yet flexible enough to yield high-quality state estimates. The resulting Bellman filter contains the Kalman filter including its extended and iterated versions (e.g. Anderson and Moore, 2012) as special cases. Like the Kalman filter, the Bellman filter is computationally inexpensive. Unlike the Kalman filter, it is driven by the *score* of the observation density rather than the prediction error. This makes the Bellman filter robust when faced with heavy-tailed observation noise, and (as we show in section 6) applicable to a substantially wider class of models than local-level models, e.g. involving count data, intensity, duration, volatility and correlation.

To circumvent the second drawback of the mode — the inability to generate parameter estimates — computationally intensive (Monte Carlo) methods have been considered by many authors (Durbin and Koopman, 2000, 2002; Jungbacker and Koopman, 2007; Richard and Zhang, 2007; Koopman et al., 2015, 2016, to name a few). To achieve computational simplicity, we deviate from this strand of literature by numerically maximising the approximate log likelihood implied by the Bellman filter. We decompose the log likelihood into (a) the ‘fit’ of the Bellman-filtered states in view of the data, minus (b) the realised Kullback-Leibler (KL, see Kullback and Leibler, 1951) divergence between filtered and predicted state densities. Intuitively, we wish to maximise the congruence between Bellman-filtered states and the data, while minimising the distance between the filtered and predicted states. All parts of the decomposition are given, or can be approximated, by the output of the Bellman filter. This has the advantage that standard gradient-based numerical optimisers can be used, making parameter estimation feasible and no more computationally demanding than ordinary estimation of the Kalman filter using maximum likelihood.

We study the performance of the Bellman filter in extensive simulation studies involving a wide range of data-generating processes. It turns out that we sacrifice little in terms of optimality when compared to the posterior mode: the mean absolute error (MAE) of the Bellman-filtered states is only marginally worse — at most  $\sim 3\%$  — than that of the (generally infeasible) posterior mode estimator, which is evaluated at the true parameters. This holds even when the Bellman-filtered states are computed using parameters estimated out-of-sample; i.e. in a training set distinct from the evaluation set used to compute MAEs.

## 2 Filtering using the posterior mode

The state-space model under consideration is given in equation (1). A realised path is denoted by  $\mathbf{y}_{1:t}(\omega)$  for every event  $\omega \in \Omega$ , where  $\Omega$  denotes the event space of the underlying complete probability space of interest, denoted  $(\Omega, \mathcal{F}, \mathbb{P})$ . We continue to use generic notation in that we write the logarithm of joint and conditional probability densities as  $\ell(\cdot, \cdot) := \log p(\cdot, \cdot)$  and  $\ell(\cdot|\cdot) := \log p(\cdot|\cdot)$ , respectively, for potentially different densities  $p$ . Here we focus on the filtering problem; any dependence on  $\psi$  is suppressed.

The joint log-likelihood function of the states and the data is written as  $\ell(\mathbf{a}_{1:t}, \mathbf{y}_{1:t})$ . Here, the data  $\mathbf{y}_{1:t}$  are considered fixed and known, while the states  $\mathbf{a}_{1:t}$  in Roman font are considered variables to be evaluated along any path  $\mathbf{a}_{1:t} \in \mathbb{R}^{m \times t}$ . For the state-space model (1), the joint log likelihood can be found by means of the ‘probability chain rule’ (Godsill et al., 2004, p. 156) as follows:

$$\ell_t(\mathbf{a}_{1:t}) := \ell(\mathbf{a}_{1:t}, \mathbf{y}_{1:t}) = \sum_{i=1}^t \ell(\mathbf{y}_i|\mathbf{a}_i) + \sum_{i=2}^t \ell(\mathbf{a}_i|\mathbf{a}_{i-1}) + \ell(\mathbf{a}_1). \quad (2)$$

On the left-hand side, we write  $\ell_t(\mathbf{a}_{1:t})$  with subscript  $t$  indicate that the joint log-likelihood function is, *a priori*, a random function of the observations  $\mathbf{y}_{1:t}$ , even though the data are considered known and fixed *ex post*. Next, the posterior mode is defined as the sequence of states that maximise equation (2).

**Definition 1 (Posterior mode)** *Assuming it exists and is unique, the posterior mode is*

$$\tilde{\mathbf{a}}_{1:t|t} := (\tilde{\mathbf{a}}_{1|t}, \tilde{\mathbf{a}}_{2|t}, \dots, \tilde{\mathbf{a}}_{t|t}) = \arg \max_{\mathbf{a}_{1:t} \in \mathbb{R}^{m \times t}} \ell_t(\mathbf{a}_{1:t}), \quad t \leq n. \quad (3)$$

Elements of the posterior mode are denoted by  $\tilde{\mathbf{a}}_{i|j}$  for  $i \leq j \leq n$ , where  $i$  denotes the state that is estimated, while  $j$  denotes the information set used. The entire solution is denoted  $\tilde{\mathbf{a}}_{1:t|t} := (\tilde{\mathbf{a}}_{1|t}, \tilde{\mathbf{a}}_{2|t}, \dots, \tilde{\mathbf{a}}_{t|t})$ , which is a collection of  $t$  vectors. To find the posterior mode for a fixed time  $t \leq n$ , iterative solution methods were proposed in Shephard and Pitt (1997), Durbin (1997) and Durbin and Koopman (2000), who use Newton’s method, and So (2003), who uses quadratic hill climbing.

While the number of state vectors to be estimated increases linearly with time, required computing times typically grow exponentially. This is unfortunate because, for the purposes of online filtering, we are predominantly interested in the last element of  $\tilde{\mathbf{a}}_{1:t|t}$ , i.e.  $\tilde{\mathbf{a}}_{t|t}$ , but for all times  $t \leq n$ . To obtain the desired sequence of real-time filtered states  $\{\tilde{\mathbf{a}}_{t|t}\}_{t=1, \dots, n}$ , we have to compute the mode  $\tilde{\mathbf{a}}_{1:t|t}$  for all times  $t$ , and extract the final element for each time  $t$  as our optimal filtered

state estimate. Each filtered state estimate  $\tilde{\mathbf{a}}_{t|t}$  thus requires the (increasingly large) optimisation problem (3) to be solved.

**Definition 2 (Optimal but infeasible filter)** *Optimal but generally infeasible filtered state estimates are*

$$\tilde{\mathbf{a}}_{t_0|t_0}, \dots, \tilde{\mathbf{a}}_{t|t}, \dots, \tilde{\mathbf{a}}_{n|n}, \quad (4)$$

where  $1 \leq t_0 \leq n$  is large enough to ensure that  $\tilde{\mathbf{a}}_{t_0|t_0}$  exists.

Estimator (4) is viewed here as an optimal estimator that is generally infeasible, because it is computed using the true (hyper)parameter  $\psi$ .

The above discussion raises the question whether it is possible, even in principle, to proceed in real time without computing a large and increasing number of ‘smoothed’ state estimates. As we show in the next section, this is indeed possible when we make use of Bellman’s dynamic programming principle.

### 3 Bellman filter

In this section our focus remains on the filtering problem; the (hyper)parameter  $\psi$  is considered given. (The estimation problem is addressed in section 5.) The Bellman-filtered states are denoted  $\{\mathbf{a}_{t|t}\}$  and will be shown to coincide with those of the Kalman filter if model (1) is linear and Gaussian. To emphasise this equality, our notation is analogous to that used by Harvey (1990) and Durbin and Koopman (2000) for the Kalman filter.

To understand how a recursive approach may be feasible, we start by noting that the joint log-likelihood function (2) satisfies a straightforward recursive relation for  $2 \leq t \leq n$  as follows:

$$\ell_t(\mathbf{a}_{1:t}) = \ell(\mathbf{y}_t|\mathbf{a}_t) + \ell(\mathbf{a}_t|\mathbf{a}_{t-1}) + \ell_{t-1}(\mathbf{a}_{1:t-1}). \quad (5)$$

That is, in transitioning from time  $t - 1$  to time  $t$ , two terms are added: one representing the state-transition density,  $\ell(\mathbf{a}_t|\mathbf{a}_{t-1})$ ; the other representing the observation density,  $\ell(\mathbf{y}_t|\mathbf{a}_t)$ . Scott (2002, p. 339) called a related equation the ‘likelihood recursion’; in the same vein, equation (5) could be dubbed the ‘log-likelihood recursion’.

Next, using standard terminology from the dynamic programming (DP) literature, we define the *value function* by maximising  $\ell_t(\mathbf{a}_{1:t})$  with respect to all states apart from the most recent state  $\mathbf{a}_t$ .

**Definition 3 (Value function)** *The value function  $V_t : \Omega \times \mathbb{R}^m \rightarrow \mathbb{R}$  is*

$$V_t(\mathbf{a}_t) := \max_{\mathbf{a}_{1:t-1} \in \mathbb{R}^{m \times (t-1)}} \ell_t(\mathbf{a}_{1:t}). \quad (6)$$

The value function  $V_t(\mathbf{a}_t)$  can be viewed as the log-likelihood function of the state  $\mathbf{a}_t$  conditional on the data  $\mathbf{y}_{1:t}$ , where all previous states have been ‘maximised out’. As such,  $V_t(\mathbf{a}_t)$  depends on the data  $\mathbf{y}_{1:t}$ , as indicated by the subscript  $t$ , which are considered fixed, as well as on its argument  $\mathbf{a}_t$ , which is considered a variable. The recursive structure (5) implies that the value function (6) satisfies Bellman’s equation as stated in the next result.

**Proposition 1 (Bellman equation)** *Suppose  $\tilde{\mathbf{a}}_{t|t}$  exists for all  $t \geq t_0$ , where  $1 \leq t_0 \leq n$ . The value function (6) satisfies Bellman’s equation:*

$$V_t(\mathbf{a}_t) = \ell(\mathbf{y}_t|\mathbf{a}_t) + \max_{\mathbf{a}_{t-1} \in \mathbb{R}^m} \left\{ \ell(\mathbf{a}_t|\mathbf{a}_{t-1}) + V_{t-1}(\mathbf{a}_{t-1}) \right\}, \quad t_0 < t \leq n. \quad (7)$$

Further, the Bellman-filtered states, defined as

$$\mathbf{a}_{t|t} := \arg \max_{\mathbf{a}_t \in \mathbb{R}^m} V_t(\mathbf{a}_t), \quad t_0 \leq t \leq n, \quad (8)$$

satisfy  $\mathbf{a}_{t|t} = \tilde{\mathbf{a}}_{t|t}$  for all  $t_0 \leq t \leq n$ .

Equation (7) is central to our approach; it is a Bellman equation recursively relating two value functions (Bellman, 1957). In DP terms, the optimisation over  $\mathbf{a}_{t-1}$  determines the ‘policy function’. Bellman equation (7) relates the value function  $V_t(\mathbf{a}_t)$  to the (previous) value function  $V_{t-1}(\mathbf{a}_{t-1})$  by adding one term reflecting the state transition,  $\ell(\mathbf{a}_t|\mathbf{a}_{t-1})$ ; one term reflecting the observation density,  $\ell(\mathbf{y}_t|\mathbf{a}_t)$ ; and a subsequent maximisation over a single state variable,  $\mathbf{a}_{t-1}$ . The resulting value function,  $V_t(\mathbf{a}_t)$ , no longer explicitly depends on the previous state  $\mathbf{a}_{t-1}$ , which has been ‘maximised out’, but it does depend on  $\mathbf{y}_t$  and  $\mathbf{a}_t$ . Further, it still depends on the data  $\mathbf{y}_1$  through  $\mathbf{y}_{t-1}$ , but only indirectly, i.e. through the (previous) value function  $V_{t-1}(\mathbf{a}_{t-1})$ .

While the structure of Bellman’s equation (7) holds few surprises, it has, to the best of our knowledge, not previously been derived in the context of model (1). Its apparent novelty notwithstanding, the proof of Bellman’s equation (7) follows from standard DP arguments (see Appendix A). Apart from assuming the existence of the posterior mode, no (additional) assumptions regarding  $\ell(\mathbf{y}_t|\mathbf{a}_t)$  or  $\ell(\mathbf{a}_t|\mathbf{a}_{t-1})$  are required. As such, Bellman’s equation (7) is of quite general applicability.

The Bellman equation (7) may be useful in the following sense. If the researcher has previously computed and stored the value function  $V_{t-1}(\mathbf{a}_{t-1})$ , as soon as she receives the observation  $\mathbf{y}_t$ , the Bellman equation (7) fully determines the current value function,  $V_t(\mathbf{a}_t)$ . While the value function  $V_t(\mathbf{a}_t)$  is random *a priori*, i.e. before receiving  $\mathbf{y}_t$ , it is deterministic *ex post*, i.e. after receiving  $\mathbf{y}_t$ . Thus, as the researcher receives the data  $\mathbf{y}_1$  through  $\mathbf{y}_t$ , she can iteratively compute the sequence of value functions. In turn, the sequence of value functions implies the sequence of filtered state estimates (8). In principle, therefore, Bellman’s equation (7) allows us to recursively compute a sequence of filtered estimates without computing a large and increasing number of ‘smoothed’ state estimates.

The first term on the right hand-side of Bellman’s equation (7), i.e.  $\ell(\mathbf{y}_t|\mathbf{a}_t)$ , implies that the score  $d\ell(\mathbf{y}_t|\mathbf{a}_t)/d\mathbf{a}_t$  will play an important role in computing the filtered state (8). For heavy-tailed observation densities, the score is typically more robust than the prediction error, which is used in the Kalman filter. This appearance of the score gives the Bellman filter its robustness, as demonstrated further in simulation studies (see section 6).

As it turns out, Bellman’s equation (7) can be solved in closed form, for all time steps, if model (1) is linear and Gaussian. In this case, the observation law  $\ell(\mathbf{y}_t|\mathbf{a}_t)$  is multivariate quadratic in  $\mathbf{a}_t$ , while the state-transition law  $\ell(\mathbf{a}_t|\mathbf{a}_{t-1})$  is multivariate quadratic in terms of both  $\mathbf{a}_t$  and  $\mathbf{a}_{t-1}$ . Finally, if the researcher’s knowledge of the previous state  $\mathbf{a}_{t-1}$  is Gaussian, then the value function  $V_{t-1}(\mathbf{a}_{t-1})$ , being a quantity in log space, is also multivariate quadratic. In this case, all optimisations required in equations (7) and (8) can be performed in closed form. The new value



function is also quadratic, but with adjusted parameters. We then obtain the Kalman filter as highlighted by the next result, the proof of which is contained in the next section.

**Corollary 1 (Kalman filter as a special case)** *Take a linear Gaussian state-space model with observation equation  $\mathbf{y}_t = \mathbf{d} + \mathbf{Z} \boldsymbol{\alpha}_t + \boldsymbol{\varepsilon}_t$ , where  $\boldsymbol{\varepsilon}_t \sim \text{i.i.d. } \mathbf{N}(\mathbf{0}, \mathbf{H})$ , and state-transition equation  $\boldsymbol{\alpha}_t = \mathbf{c} + \mathbf{T} \boldsymbol{\alpha}_{t-1} + \boldsymbol{\eta}_t$ , where  $\boldsymbol{\eta}_t \sim \text{i.i.d. } \mathbf{N}(\mathbf{0}, \mathbf{Q})$ , such that Kalman’s (1960) filter applies. Assume the Kalman-filtered covariance matrices  $\{\mathbf{P}_{t|t}\}$  are positive definite. Then the Bellman-filtered states  $\{\boldsymbol{\alpha}_{t|t}\}$  are identical to the Kalman-filtered states and the value function (6) is multivariate quadratic for every time step with a negative Hessian equal to  $\mathbf{I}_{t|t} = \mathbf{P}_{t|t}^{-1}$ .*

In general, Bellman’s equation (7) cannot be solved in closed form. Hence, the price paid for reducing the number of optimisations for each time step is that we must derive and store the (potentially complicated) value function  $V_t(\boldsymbol{\alpha}_t)$  for each time step. To approximate it, we could sample the function  $V_t(\boldsymbol{\alpha}_t)$  at a large number of points and store it non-parametrically for each time step. This would be especially attractive if the hidden state  $\boldsymbol{\alpha}_t$  took values in a finite subset of  $\mathbb{R}^m$ , in which case it would be obvious which points in  $\mathbb{R}^m$  to sample. For such models, this line of reasoning leads to the Viterbi algorithm developed for speech recognition (e.g. Levinson et al., 1983; Godsill et al., 2001, p. 86), which is then a special case of the Bellman filter. In the context of states taking continuous values in  $\mathbb{R}^m$ , however, it is more natural to approximate value functions using polynomials (e.g. Weierstrass, 1885). Using either method (sampling or fitting polynomials), we could subsequently define approximate state estimates by computing the argmax of approximate value functions, in analogy with the argmax (8). As the approximation of value functions grows in accuracy, we should be able to recover the state estimates implied by the (exact) Bellman equation (7).

Although it is possible to develop asymptotic theory for approximating value functions using polynomials, doing so seems disproportionate relative to our aims. Indeed, a simpler approach will be shown to be sufficiently accurate. Specifically, motivated by Corollary 1 above, we propose to approximate the value function for each time step using a multivariate quadratic function, which is parametrised by the argmax and the matrix of second derivatives at the peak. While generally inexact, multivariate quadratic functions can still approximate smooth value functions around their global maximum. The simulation results in section 6 are so compelling that considering approximation methods more sophisticated than fitting quadratic functions appears to be unnecessary, at least for applications in time-series econometrics.

## 4 Bellman filter for models with linear Gaussian state dynamics

The general applicability of Bellman’s equation (7) notwithstanding, this section focuses on the filtering problem for models in which the state-transition equation remains linear and Gaussian. Such models are written as

$$\mathbf{y}_t \sim p^\psi(\mathbf{y}_t | \boldsymbol{\alpha}_t), \quad \boldsymbol{\alpha}_t = \mathbf{c} + \mathbf{T} \boldsymbol{\alpha}_{t-1} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \text{i.i.d. } \mathbf{N}(\mathbf{0}, \mathbf{Q}), \quad \boldsymbol{\alpha}_1 \sim p(\boldsymbol{\alpha}_1), \quad (9)$$

where  $t = 1, \dots, n$ . The system vector  $\mathbf{c}$  and system matrix  $\mathbf{T}$ , which are assumed to be of appropriate dimensions, may depend on the fixed (hyper)parameter  $\psi$ . The covariance matrix  $\mathbf{Q}$  is assumed symmetric and positive semi-definite. The observation density  $p(\mathbf{y}_t | \boldsymbol{\alpha}_t)$  may still be

non-Gaussian and involve nonlinearity. We may employ exponential link functions to ensure that variables such as intensity or volatility remain positive. In our notation, the link function is left implicit; the observation density  $p(\mathbf{y}_t|\boldsymbol{\alpha}_t)$  may contain any desired (nonlinear) dependence on the state  $\boldsymbol{\alpha}_t$ .

We propose to use approximate dynamic programming (ADP, e.g. Bertsekas, 2012) in that we apply a multivariate quadratic approximation of the value function for each time step. The end result of the computation below is the Bellman filter stated in Table 1, which is a generalisation of the Kalman filter. Arguably, Table 1 in conjunction with the proposed estimation method (see section 5) represents our main contribution for practitioners.

To derive the Bellman filter for model (9), we start with the (exact) Bellman equation (7). In practice, the behaviour of  $V_{t-1}(\mathbf{a})$  around its peak turns out to be most relevant for the determination of the next value function,  $V_t(\mathbf{a})$ . The value function  $V_{t-1}(\mathbf{a})$  could be approximated locally, around its peak, by a multivariate quadratic function with two parameters, being the argmax and the matrix of second derivatives at the peak, i.e.

$$V_{t-1}(\mathbf{a}_{t-1}) \approx -\frac{1}{2}(\mathbf{a}_{t-1} - \mathbf{a}_{t-1|t-1})' \mathbf{I}_{t-1|t-1} (\mathbf{a}_{t-1} - \mathbf{a}_{t-1|t-1}) + \text{constants}, \quad (10)$$

for some state estimate  $\mathbf{a}_{t-1|t-1}$  and precision matrix  $\mathbf{I}_{t-1|t-1}$ , which is assumed to be symmetric and positive definite. Here, the state  $\mathbf{a}_{t-1}$  is considered a variable, while  $\mathbf{a}_{t-1|t-1}$  is an estimate. Constants can be ignored, as we are interested only in the location of the maximum and the sharpness of the peak, not in the height of the value function. Approximation (10) would be justified not only locally but also globally if our knowledge at time  $t-1$  of the state  $\boldsymbol{\alpha}_{t-1}$  were accurately described by a normal distribution with mean  $\mathbf{a}_{t-1|t-1}$  and precision matrix  $\mathbf{I}_{t-1|t-1}$ . If the model is such that the Kalman filter applies, the approximation (10) thus happens to be exact. Moreover, approximation (10) is exact for  $t=1$  if the model is stationary and  $\boldsymbol{\alpha}_0$  is drawn from the states' unconditional distribution, which is also multivariate normal (e.g. Harvey, 1990, p. 121). The initialisation of the Bellman filter based on the unconditional distribution is indicated in Table 1 under the step 'Initialise'.

The state transition in model (9) is linear and Gaussian, such that  $\ell(\mathbf{a}_t|\mathbf{a}_{t-1})$  is a quadratic function of both state variables as follows:

$$\ell(\mathbf{a}_t|\mathbf{a}_{t-1}) = -\frac{1}{2}(\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1})' \mathbf{Q}^{-1} (\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1}) + \text{constants}, \quad (11)$$

where  $\mathbf{a}_t$  and  $\mathbf{a}_{t-1}$  are variables in  $\mathbb{R}^m$ . (If  $\mathbf{Q}$  is only positive semi-definite, its inverse can be interpreted in a generalised sense.) Next, substituting the quadratic approximation (10) and the exact state transition (11) into Bellman's equation (7), we obtain

$$\begin{aligned} V_t(\mathbf{a}_t) = \ell(\mathbf{y}_t|\mathbf{a}_t) + \max_{\mathbf{a}_{t-1} \in \mathbb{R}^m} \left\{ -\frac{1}{2}(\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1})' \mathbf{Q}^{-1} (\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1}) \right. \\ \left. -\frac{1}{2}(\mathbf{a}_{t-1} - \mathbf{a}_{t-1|t-1})' \mathbf{I}_{t-1|t-1} (\mathbf{a}_{t-1} - \mathbf{a}_{t-1|t-1}) \right\} \\ + \text{constants}, \end{aligned} \quad (12)$$

which for the purposes of simplicity we write with equality, which is unproblematic as long as we keep in mind that the resulting value function is generally inexact. Conveniently, the variable  $\mathbf{a}_{t-1}$  appears at most quadratically on the right-hand side of equation (12). As such, the corresponding

Table 1: Bellman filter for model (9): A generalisation of the Kalman filter.

Step	Method	Computation
Initialise	Unconditional distribution	Set $\mathbf{a}_{0 0} = (\mathbf{1} - \mathbf{T})^{-1} \mathbf{c}$ and $\text{vec}(\mathbf{I}_{0 0}^{-1}) = (\mathbf{1} - \mathbf{T} \otimes \mathbf{T})^{-1} \text{vec}(\mathbf{Q})$ . Set $t = 1$ .
	Diffuse prior	Set $\mathbf{a}_{0 0} = \mathbf{0}$ and $\mathbf{I}_{0 0} = \mathbf{0}$ . Set $t = 1$ .
Predict		$\mathbf{a}_{t t-1} = \mathbf{c} + \mathbf{T} \mathbf{a}_{t-1 t-1}$ $\mathbf{I}_{t t-1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1} \mathbf{T} (\mathbf{I}_{t-1 t-1} + \mathbf{T}' \mathbf{Q}^{-1} \mathbf{T})^{-1} \mathbf{T}' \mathbf{Q}^{-1}$
Start		Set $k = 0$ . Set $\mathbf{a}_{t t}^{(k)} = \mathbf{a}_{t t-1}$ . Alternatively, set $\mathbf{a}_{t t}^{(k)} = \arg \max_{\mathbf{a}} \ell(\mathbf{y}_t   \mathbf{a})$ if this quantity exists.
Optimise	Newton	$\mathbf{a}_{t t}^{(k+1)} = \mathbf{a}_{t t}^{(k)} + \left\{ \mathbf{I}_{t t-1} - \frac{d^2 \ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \right\}^{-1} \left\{ \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}} - \mathbf{I}_{t t-1} (\mathbf{a} - \mathbf{a}_{t t-1}) \right\} \Bigg _{\mathbf{a}=\mathbf{a}_{t t}^{(k)}}$
	Fisher	$\mathbf{a}_{t t}^{(k+1)} = \mathbf{a}_{t t}^{(k)} + \left\{ \mathbf{I}_{t t-1} - \mathbb{E} \left[ \frac{d^2 \ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \mid \mathbf{a} \right] \right\}^{-1} \left\{ \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}} - \mathbf{I}_{t t-1} (\mathbf{a} - \mathbf{a}_{t t-1}) \right\} \Bigg _{\mathbf{a}=\mathbf{a}_{t t}^{(k)}}$
	BHHH	$\mathbf{a}_{t t}^{(k+1)} = \mathbf{a}_{t t}^{(k)} + \left\{ \mathbf{I}_{t t-1} + \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}} \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}'} \right\}^{-1} \left\{ \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}} - \mathbf{I}_{t t-1} (\mathbf{a} - \mathbf{a}_{t t-1}) \right\} \Bigg _{\mathbf{a}=\mathbf{a}_{t t}^{(k)}}$
		Set $k = k + 1$ and repeat the ‘Optimise’ step.
Stop		Stop at $k = k_{\max}$ if some convergence criterion (e.g. that we stop after a pre-specified number of iterations $k_{\max}$ ) is satisfied.
Update		$\mathbf{a}_{t t} = \mathbf{a}_{t t}^{(k_{\max})}$
	Newton	$\mathbf{I}_{t t} = \mathbf{I}_{t t-1} - \frac{d^2 \ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \Bigg _{\mathbf{a}=\mathbf{a}_{t t}}$
	Fisher	$\mathbf{I}_{t t} = \mathbf{I}_{t t-1} - \mathbb{E} \left[ \frac{d^2 \ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \mid \mathbf{a} \right] \Bigg _{\mathbf{a}=\mathbf{a}_{t t}}$
	BHHH	$\mathbf{I}_{t t} = \mathbf{I}_{t t-1} + \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}} \frac{d\ell(\mathbf{y}_t   \mathbf{a})}{d\mathbf{a}'} \Bigg _{\mathbf{a}=\mathbf{a}_{t t}}$
Proceed		Set $t = t + 1$ and return to the step ‘Predict’.

*Note:* The log-likelihood function  $\ell(\mathbf{y}_t | \boldsymbol{\alpha}_t)$  is known in closed form and can be read off from the data-generating process (9). The corresponding score and the realised and expected information quantities are written as  $d\ell(\mathbf{y}_t | \mathbf{a})/d\mathbf{a}$ ,  $-d^2 \ell(\mathbf{y}_t | \mathbf{a})/(d\mathbf{a} d\mathbf{a}')$  and  $\mathbb{E}[-d^2 \ell(\mathbf{y}_t | \mathbf{a})/(d\mathbf{a} d\mathbf{a}')] | \mathbf{a}$ , respectively, which are viewed as functions of  $\mathbf{a}$ , to be evaluated at some state estimate. Under the steps ‘Optimise’ and ‘Update’, researchers have a choice of three methods, which may but need not be identical for both steps.

maximisation can be performed in closed form. Because the maximisation is still based on a quadratic approximation of the previous value function, the resulting argmax should be viewed as a ‘closed-form approximation’; in our notation, we ignore this subtlety. Computing the first-order condition in equation (12) and solving for  $\mathbf{a}_{t-1}$ , we obtain

$$\mathbf{a}_{t-1}^* = (\mathbf{I}_{t-1|t-1} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T})^{-1} \{ \mathbf{I}_{t-1|t-1} \mathbf{a}_{t-1|t-1} + \mathbf{T}'\mathbf{Q}^{-1}(\mathbf{a}_t - \mathbf{c}) \}. \quad (13)$$

The solution  $\mathbf{a}_{t-1}^*$  depends linearly on  $\mathbf{a}_t$ , i.e. we could have written  $\mathbf{a}_{t-1}^* = \mathbf{a}_{t-1}^*(\mathbf{a}_t)$ ; this dependence shall henceforth be important.

Substituting the argmax (13) back into equation (12), which was to be optimised, and performing some algebra (for details, see Appendix B), we find

$$V_t(\mathbf{a}_t) = \ell(\mathbf{y}_t|\mathbf{a}_t) - \frac{1}{2}(\mathbf{a}_t - \mathbf{a}_{t|t-1})' \mathbf{I}_{t|t-1}(\mathbf{a}_t - \mathbf{a}_{t|t-1}) + \text{constants}, \quad (14)$$

where  $\mathbf{a}_t$  is the only remaining variable on the right-hand side, and we have defined the predicted state  $\mathbf{a}_{t|t-1}$  and predicted precision matrix  $\mathbf{I}_{t|t-1}$  as follows:

$$\mathbf{a}_{t|t-1} := \mathbf{c} + \mathbf{T} \mathbf{a}_{t-1|t-1}, \quad (15)$$

$$\mathbf{I}_{t|t-1} := \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{T}(\mathbf{I}_{t-1|t-1} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T})^{-1}\mathbf{T}'\mathbf{Q}^{-1}. \quad (16)$$

Equations (15) and (16) are collected in Table 1 under the step ‘Predict’. As equation (14) indicates, we are left with the log likelihood of a single observation,  $\ell(\mathbf{y}_t|\mathbf{a}_t)$ , and a quadratic term centred at the prediction  $\mathbf{a}_{t|t-1}$ , and with precision  $\mathbf{I}_{t|t-1}$ .

While our derivation is different, the resulting equations (15) and (16) turn out to be identical to the prediction steps of the Kalman filter. For equation (15), this is obvious; see e.g. Harvey (1990, p. 106). For equation (16), the relationship with Kalman’s prediction step is somewhat obscured because it is written in the information rather than the covariance form. To clarify, this, suppose that the inverses  $\mathbf{P}_{t-1|t-1} := \mathbf{I}_{t-1|t-1}^{-1}$  and  $\mathbf{P}_{t|t-1} := \mathbf{I}_{t|t-1}^{-1}$  exist. Following from the Woodbury matrix identity (e.g. Henderson and Searle, 1981, eq. 1), equation (16) implies  $\mathbf{P}_{t|t-1} = \mathbf{T} \mathbf{P}_{t-1|t-1} \mathbf{T}' + \mathbf{Q}$ . This expression is immediately recognisable as the covariance matrix prediction step of the Kalman filter (Harvey, 1990, p. 106).

While predictions (15) and (16) turned out to be identical to those of the (information form of the) Kalman filter, the updating equations, derived next, turn out to be different in general, while still containing those of the Kalman filter as a special case. Taking approximate value function (14) as given, we may take the updated state estimate  $\mathbf{a}_{t|t}$  and the updated precision matrix  $\mathbf{I}_{t|t}$  to be

$$\mathbf{a}_{t|t} = \operatorname{argmax}_{\mathbf{a} \in \mathbb{R}^m} V_t(\mathbf{a}), \quad \mathbf{I}_{t|t} = - \left. \frac{d^2 V_t(\mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \right|_{\mathbf{a}=\mathbf{a}_{t|t}}. \quad (17)$$

The argmax determines our filtered state estimate, while the computation of second derivatives at the peak facilitates the recursive approach, where each value function is approximated quadratically around its peak. The expression for the updated information matrix  $\mathbf{I}_{t|t}$  is ‘local’ in the sense that it computes second derivatives at a single point; global fitting methods could also be used.

For the value function  $V_t(\mathbf{a})$  in equation (14) to possess a unique global optimum, it is sufficient that the matrix of second derivatives, i.e.  $d^2\ell(\mathbf{y}_t|\mathbf{a})/(d\mathbf{a}d\mathbf{a}') - \mathbf{I}_{t|t-1}$ , is negative definite for all  $\mathbf{a} \in \mathbb{R}^m$ , where  $-d^2\ell(\mathbf{y}_t|\mathbf{a})/(d\mathbf{a}d\mathbf{a}')$  is the *realised marginal information*. Even if the existence

of a global maximum is guaranteed, the potentially complicated functional form  $\ell(\mathbf{y}_t|\mathbf{a}_t)$  implies that the maximisation over  $\mathbf{a}_t$  in equation (14) cannot, in general, be performed analytically. The obvious exception is when the model is linear and Gaussian, in which case  $\ell(\mathbf{y}_t|\mathbf{a}_t)$  is quadratic in  $\mathbf{a}_t$ , such that the entire objective function (14) is quadratic in  $\mathbf{a}_t$ . The updated state  $\mathbf{a}_{t|t}$  can then be found in closed form, and, as shown below, is equivalent to the Kalman-filtered update.

In general, we may write down analytically the steps of Newton’s optimisation method (e.g. Nocedal and Wright, 2006). Indeed, a plain-vanilla application of Newton’s method to maximising  $V_t(\mathbf{a})$  with respect to  $\mathbf{a}$  would read

$$\mathbf{a}_{t|t}^{(k+1)} = \mathbf{a}_{t|t}^{(k)} + \left[ -\frac{d^2 V_{t+1}(\mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \right]^{-1} \frac{dV_t(\mathbf{a})}{d\mathbf{a}} \Bigg|_{\mathbf{a}=\mathbf{a}_{t|t}^{(k)}}, \quad (18)$$

where elements of the resulting sequence are denoted as  $\mathbf{a}_{t|t}^{(k)}$  for  $k = 0, 1, \dots$ . As indicated in Table 1 under the step ‘Start’, Newton’s method (18) requires an initialisation to be specified, e.g.  $\mathbf{a}_{t|t}^{(0)} = \mathbf{a}_{t|t-1}$ , such that the starting point for the optimisation is equal to the predicted state.

Recalling value function (14), the gradient and negative Hessian required by Newton’s optimisation method (18) can be approximated in closed form as follows:

$$\frac{dV_t(\mathbf{a})}{d\mathbf{a}} = \frac{d\ell(\mathbf{y}_t|\mathbf{a})}{d\mathbf{a}} - \mathbf{I}_{t|t-1}(\mathbf{a} - \mathbf{a}_{t|t-1}), \quad (19)$$

$$-\frac{d^2 V_t(\mathbf{a})}{d\mathbf{a} d\mathbf{a}'} = \mathbf{I}_{t|t-1} - \frac{d^2 \ell(\mathbf{y}_t|\mathbf{a})}{d\mathbf{a} d\mathbf{a}'}. \quad (20)$$

As the observation  $\mathbf{y}_t$  is fixed, the score in equation (19) and the realised marginal information in equation (20) are viewed as functions of the state variable  $\mathbf{a}$ . Simply put, the Bellman filter is obtained by substituting the gradient (19) and Hessian (20) into Newton’s method (18). The resulting iterative Newton method is stated in Table 1 under the step ‘Optimise’. The presence of the score in the optimisation step is distinctive for the Bellman filter and guarantees its robustness if the observation density is heavy tailed. As indicated under the steps ‘Stop’ and ‘Update’, we may perform a fixed number of Newton steps, or as many as are required according to some convergence criterion, after which we set the final estimate  $\mathbf{a}_{t|t}$  equal to  $\mathbf{a}_{t|t}^{(k_{\max})}$ , where  $k_{\max}$  is the number of iterations performed.

Regarding the updated precision matrix  $\mathbf{I}_{t|t}$ , we may evaluate (the negative of) the Hessian (20) at the peak, as indicated under Newton’s method under the step ‘Update’ in Table 1. After the updating step, we set  $t = t + 1$  and return to the prediction step, as indicated in Table 1 under the step ‘Proceed’.

Newton’s method is applicable if the scaling matrix  $\mathbf{I}_{t|t-1} - d^2 \ell(\mathbf{y}_t|\mathbf{a})/(d\mathbf{a}d\mathbf{a}')$  is positive definite. If the realised marginal information  $-d^2 \ell(\mathbf{y}_t|\mathbf{a})/(d\mathbf{a}d\mathbf{a}')$  fails to be positive semi-definite for some values of  $\mathbf{y}_t \in \mathbb{R}^l$  and  $\mathbf{a} \in \mathbb{R}^m$ , we may, in order to guarantee well-defined optimisation steps, resort to Fisher’s scoring method or the Berndt-Hall-Hall-Hausman (BHHH) algorithm, which are also given in Table 1. These optimisation methods differ from Newton’s method in their approximation of the Hessian matrix and suggest slightly different approximations for the updated precision matrix  $\mathbf{I}_{t|t}$ , as indicated under the step ‘Update’ in Table 1 (see Appendix C for further discussion). Generalising to more sophisticated optimisation methods, e.g. the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, is straightforward but, for applications in time-series econometrics, rarely needed.

## 4.1 Kalman filter and its extensions as a special case of the Bellman filter

To see how the Bellman filter in Table 1 relates to the Kalman filter (KF), we consider the linear Gaussian state-space model stated in Corollary 1. Suppose the inverse of the Kalman-filtered covariance matrix exists, i.e.  $\mathbf{P}_{t-1|t-1}^{-1} := \mathbf{I}_{t-1|t-1}$  exists, such that the value function at time  $t-1$  can be written as in equation (10), which is then exact. For our optimisation, we take the starting point  $\mathbf{a}_{t|t}^{(0)} = \mathbf{a}_{t|t-1}$ , and use Newton or Fisher optimisation steps. For a Gaussian observation density with level  $\mathbf{d} + \mathbf{Z} \mathbf{a}_t$ , the log likelihood  $\ell(\mathbf{y}_t|\mathbf{a}_t)$  is multivariate quadratic in  $\mathbf{a}_t$ , such that the entire objective function (14) turns out to be multivariate quadratic in  $\mathbf{a}_t$ . The matrix of second derivatives is constant, such that Newton and Fisher optimisation steps are identical. Moreover, given the quadratic nature of the objective function, both methods find the location of the optimum in a single step. Indeed, the result is the classic Kalman filter, albeit written in the information form (for details, see Appendix D). Viewed in this light, the single-step update of the Kalman filter is an anomaly brought about by the exact quadratic nature of the objective function (14).

To see how the Bellman filter in Table 1 relates to the iterated extended Kalman filter (IEKF), we take  $\mathbf{y}_t = \mathbf{d} + \mathbf{Z}(\mathbf{a}_t) + \boldsymbol{\varepsilon}_t$  for some nonlinear vector function  $\mathbf{Z}(\cdot)$  and  $\boldsymbol{\varepsilon}_t \sim \text{i.i.d. } \mathbf{N}(\mathbf{0}, \mathbf{H})$  as before. Again, we take the starting point  $\mathbf{a}_{t|t}^{(0)} = \mathbf{a}_{t|t-1}$  and perform Fisher optimisation steps, where we choose to ignore (i.e. set to zero) all second-order derivatives of  $\mathbf{Z}(\cdot)$ . We then obtain the IEKF as a special case (for details, see Appendix E).

In sum, both the Kalman filter and its extended and iterated versions fall under the umbrella of the Bellman filter. If the observation noise  $\boldsymbol{\varepsilon}_t$  is heavy tailed, however, the Bellman filter in Table 1 suggests a ‘robustified’ version of the KF and IEKF, in which case the tail behaviour of  $p(\mathbf{y}_t|\mathbf{a}_t)$  is accounted for in the optimisation step by using the score  $d\ell(\mathbf{y}_t|\mathbf{a}_t)/d\mathbf{a}_t$ .

## 5 Estimation method

This section considers the estimation problem, as distinct from the filtering problem, in that we aim to estimate both the time-varying states  $\boldsymbol{\alpha}_{1:t}$  and the constant (hyper)parameter  $\boldsymbol{\psi}$ . As before, we take model (9) with linear Gaussian state dynamics, and we continue to assume the existence of the posterior mode.

To estimate the constant parameter  $\boldsymbol{\psi}$ , computationally intensive methods have been considered by many authors (see section 1). We deviate from this strand of literature by decomposing the log likelihood in terms of the ‘fit’ generated by the Bellman filter, penalised, roughly speaking, by the realised Kullback-Leibler (KL, see Kullback and Leibler, 1951) divergence between filtered and predicted state densities. The proposed decomposition has the advantage that all terms can be evaluated or approximated using the output of the Bellman filter in Table 1; no sampling techniques or numerical integration methods are required. As a result, the estimation method is no more time consuming or computationally intensive than ordinary estimation of the Kalman filter using maximum likelihood.

To introduce the proposed decomposition, we focus on the log-likelihood contribution of a single observation,  $\ell(\mathbf{y}_t|\mathcal{F}_{t-1}) := \log p(\mathbf{y}_t|\mathcal{F}_{t-1})$ . The next computation is straightforward and holds for

all  $\mathbf{y}_t \in \mathbb{R}^l$  and all  $\boldsymbol{\alpha}_t \in \mathbb{R}^m$ :

$$\begin{aligned} \ell(\mathbf{y}_t|\mathcal{F}_{t-1}) &= \ell(\mathbf{y}_t, \boldsymbol{\alpha}_t|\mathcal{F}_{t-1}) - \ell(\boldsymbol{\alpha}_t|\mathbf{y}_t, \mathcal{F}_{t-1}), \\ &= \ell(\mathbf{y}_t, \boldsymbol{\alpha}_t|\mathcal{F}_{t-1}) - \ell(\boldsymbol{\alpha}_t|\mathcal{F}_t), \\ &= \ell(\mathbf{y}_t|\boldsymbol{\alpha}_t) + \ell(\boldsymbol{\alpha}_t|\mathcal{F}_{t-1}) - \ell(\boldsymbol{\alpha}_t|\mathcal{F}_t). \end{aligned} \quad (21)$$

The first line uses the laws of joint and conditional densities, while the second uses the basic fact  $\mathbf{y}_t \cup \mathcal{F}_{t-1} = \mathcal{F}_t$ . The third line is essentially the Markov property: the observation  $\mathbf{y}_t$  is conditionally independent of  $\mathcal{F}_{t-1}$  when the state  $\boldsymbol{\alpha}_t$  is given.

While the above decomposition is valid for any  $\boldsymbol{\alpha}_t \in \mathbb{R}^m$ , the resulting expression is not a computable quantity, as  $\boldsymbol{\alpha}_t$  remains unknown. In principle the expression could be evaluated at any value of  $\boldsymbol{\alpha}_t$ , but it turns out to be practical to take a specific value, namely the Bellman-filtered state estimate  $\mathbf{a}_{t|t}$ . Swapping the order of the last two terms, we obtain

$$\ell(\mathbf{y}_t|\mathcal{F}_{t-1}) = \ell(\mathbf{y}_t|\boldsymbol{\alpha}_t) \Big|_{\boldsymbol{\alpha}_t=\mathbf{a}_{t|t}} - \left\{ \ell(\boldsymbol{\alpha}_t|\mathcal{F}_t) - \ell(\boldsymbol{\alpha}_t|\mathcal{F}_{t-1}) \right\} \Big|_{\boldsymbol{\alpha}_t=\mathbf{a}_{t|t}}. \quad (22)$$

Here we have split the log likelihood corresponding to the observation  $\mathbf{y}_t$  into two parts. The first part,  $\ell(\mathbf{y}_t|\boldsymbol{\alpha}_t)$  evaluated at  $\boldsymbol{\alpha}_t = \mathbf{a}_{t|t}$ , quantifies the congruence (or ‘fit’) between the Bellman-filtered state  $\mathbf{a}_{t|t}$  and the observation  $\mathbf{y}_t$ , which we wish to maximise. We simultaneously want to minimise the realised KL divergence between predictions and updates, as determined by the difference between the two terms in curly brackets. This ‘KL penalty’ prevents our updates from deviating too drastically from our predictions. The trade-off between maximising the first term and minimising the second, which appears with a minus sign, give rise to a meaningful optimisation problem.

Despite the simplicity of our proposed ‘fit minus KL divergence’ decomposition (22), we have not been able to find it in the literature. In addition to making intuitive sense, the decomposition allows us to avoid numerical integration techniques. While the decomposition itself is exact, we do not generally have an exact expression for the KL divergence. To ensure that the log-likelihood contribution (22) is computable, we now turn to approximating the KL divergence term.

In deriving the Bellman filter for model (9), we presumed that the researcher’s knowledge, as measured in log-likelihood space for each time step, could be approximated by a multivariate quadratic function. Extending this line of reasoning, we may consider the following approximation of the two terms that compose the realised KL divergence:

$$\ell(\boldsymbol{\alpha}_t|\mathcal{F}_t) \approx \frac{1}{2} \log \det\{\mathbf{I}_{t|t}/(2\pi)\} - \frac{1}{2}(\boldsymbol{\alpha}_t - \mathbf{a}_{t|t})' \mathbf{I}_{t|t} (\boldsymbol{\alpha}_t - \mathbf{a}_{t|t}), \quad (23)$$

$$\ell(\boldsymbol{\alpha}_t|\mathcal{F}_{t-1}) \approx \frac{1}{2} \log \det\{\mathbf{I}_{t|t-1}/(2\pi)\} - \frac{1}{2}(\boldsymbol{\alpha}_t - \mathbf{a}_{t|t-1})' \mathbf{I}_{t|t-1} (\boldsymbol{\alpha}_t - \mathbf{a}_{t|t-1}). \quad (24)$$

Here the state  $\boldsymbol{\alpha}_t$  is understood as a variable, not a realisation, while  $\mathbf{a}_{t|t-1}$ ,  $\mathbf{a}_{t|t}$ ,  $\mathbf{I}_{t|t-1}$  and  $\mathbf{I}_{t|t}$  are produced by the Bellman filter in Table 1. If we evaluate the unknown state variable  $\boldsymbol{\alpha}_t$  at the computable quantity  $\mathbf{a}_{t|t}$ , we obtain a computable approximation of the KL divergence term. The quality of our parameter estimates in the next section suggests that, even though updated and predicted densities may themselves be non-normal, the KL divergence between them may resemble that between two (calibrated) normal distributions. Naturally, if the model is linear and Gaussian, then the Bellman filter is exact (it is, in fact, the Kalman filter), as are equations (23) and (24).

We propose using the (exact) ‘fit minus KL divergence’ decomposition (22) in combination with approximations (23) and (24) to write the log likelihood of the data  $\mathbf{y}_{1:n}$  given  $\boldsymbol{\psi}$  as follows:

$$\begin{aligned} \ell(\boldsymbol{\psi}) &:= \sum_{t=1}^n \ell(\mathbf{y}_t | \mathcal{F}_{t-1}) = \sum_{t=1}^n \left[ \ell(\mathbf{y}_t | \boldsymbol{\alpha}_t) - \{ \ell(\boldsymbol{\alpha}_t | \mathcal{F}_t) - \ell(\boldsymbol{\alpha}_t | \mathcal{F}_{t-1}) \} \right]_{\boldsymbol{\alpha}_t = \mathbf{a}_{t|t}}, \\ &\approx \sum_{t=1}^n \ell(\mathbf{y}_t | \mathbf{a}_{t|t}) - \frac{1}{2} \log \det(\mathbf{I}_{t|t}) + \frac{1}{2} \log \det(\mathbf{I}_{t|t-1}) \\ &\quad - \frac{1}{2} (\mathbf{a}_{t|t} - \mathbf{a}_{t|t-1})' \mathbf{I}_{t|t-1} (\mathbf{a}_{t|t} - \mathbf{a}_{t|t-1}). \end{aligned} \quad (25)$$

The first term in the sum,  $\ell(\mathbf{y}_t | \mathbf{a}_{t|t})$ , is known in closed form and determined by the observation density as specified in model (9). All other terms on the right-hand side can be computed based on the output of the Bellman filter in Table 1. Decomposition (25) can be viewed as an alternative to the prediction error decomposition for linear Gaussian state-space models (see e.g. Harvey, 1990, p. 126), the advantage being that decomposition (25) is applicable more generally.

Now that we have specified a computable (but in general approximate) log-likelihood function  $\ell(\boldsymbol{\psi})$ , a numerical gradient-based optimiser can be used for the optimisation with respect to the parameter  $\boldsymbol{\psi}$ , leading to an approximate maximum likelihood estimator (MLE) defined as

$$\begin{aligned} \hat{\boldsymbol{\psi}} &:= \arg \max_{\boldsymbol{\psi}} \left\{ \sum_{t=1}^n \ell(\mathbf{y}_t | \mathbf{a}_{t|t}) - \frac{1}{2} \log \det(\mathbf{I}_{t|t}) + \frac{1}{2} \log \det(\mathbf{I}_{t|t-1}) \right. \\ &\quad \left. - \frac{1}{2} (\mathbf{a}_{t|t} - \mathbf{a}_{t|t-1})' \mathbf{I}_{t|t-1} (\mathbf{a}_{t|t} - \mathbf{a}_{t|t-1}) \right\}, \end{aligned} \quad (26)$$

where, in general, all terms on the right-hand side implicitly or explicitly depend on the (hyper)parameter  $\boldsymbol{\psi}$ .

**Corollary 2** *Take the linear Gaussian state-space model specified in Corollary 1. Assume the Kalman-filtered covariance matrices  $\{\mathbf{P}_{t|t}\}$  are positive definite. Then estimator (26) is equal to the MLE.*

Estimator (26) is only slightly more computationally demanding than standard maximum likelihood estimation of the Kalman filter. The sole source of additional computational complexity derives from the fact that the Bellman filter in Table 1 may perform several optimisation steps for each time step, while the Kalman filter performs only one. However, because each optimisation step is straightforward and few steps are typically required, the additional computational burden is negligible.

## 6 Simulation studies

We conduct a large-scale Monte Carlo study to investigate the performance of the Bellman filter in Table 1 for a range of data-generating processes (DGPs) involving count, intensity, duration, volatility and dependence. We evaluate the performance of the Bellman filter at the true parameter  $\boldsymbol{\psi}$ , as well as at the parameter estimate  $\hat{\boldsymbol{\psi}}$  given by estimator (26), which we compute using either in-sample or out-of-sample data.



We consider 10 DGPs of type (9) with univariate states  $\{\alpha_t\}$  following linear Gaussian state dynamics. Table 2 shows 10 observation densities and link functions; jointly, equation (9) and Table 2 specify the DGPs considered in this section. The corresponding scores as well as realised and expected information quantities are presented in Table 3. To avoid selection bias on our part, Tables 2 and 3 are adapted with only minor modifications from Koopman et al. (2016). In taking the DGPs chosen by these authors, we essentially test the performance of the Bellman filter on an ‘exogenous’ set of models. We follow their set-up with three minor modifications:

1. We compute the score and information quantity by taking derivatives of  $\ell(\mathbf{y}_t|\alpha_t)$  with respect to the state  $\alpha_t \in \mathbb{R}$ , which is subject to linear Gaussian dynamics in equation (9). For us, this choice is dictated by the Bellman filter in Table 1. Conversely, Koopman et al. (2016) compute scores and information quantities by taking derivatives with respect to  $\lambda_t, \beta_t, \sigma_t^2$  or  $\rho_t$ , which are related to  $\alpha_t$  via the link functions stated in Table 2.
2. We formulate our models for dynamic dependence in terms of a joint probability density function with unit variance in both directions and a time-varying correlation coefficient  $\rho_t$ , rather than in terms of a copula as in Koopman et al. (2016). This is purely for the sake of simplicity; we do not expect this choice to have a major impact.
3. We have added one DGP to the nine considered in Koopman et al. (2016), namely a local-level model with linear Gaussian state dynamics (9) and heavy-tailed observation noise. While a local-level model with Gaussian observation noise would be solved exactly by the KF, heavy-tailed observation noise is not adjusted for in the KF. Although the KF remains the best linear unbiased estimator, it is possible that the (nonlinear) Bellman filter fares better. To investigate the effect of heavy-tailed observation noise, we use a Student’s  $t$  distribution with three degrees of freedom.

For the first seven DGPs in Table 4, the realised information is positive semi-definite (p.s.d.); we therefore run the Bellman filter in Table 1 using a fixed number of Newton optimisation steps — five — along with Newton updating steps. For both DGPs involving dependence, the realised information is not guaranteed to be p.s.d., and furthermore is unbounded above. In these cases, we use a fixed number of Fisher optimisation steps — 10, as convergence may be slower — along with Fisher updating steps. For the local-level model with heavy-tailed observation noise, the realised information may fail to be p.s.d., but is otherwise well-behaved (i.e. bounded). In this case, we use Fisher optimisation steps to iterate until convergence, and then perform Newton updating steps. Other sensible choices lead to similar results. In all cases, we initialise the Bellman filter at  $t = 0$  using the unconditional distribution, and for each optimisation step we take as our starting point the most recent prediction.

Table 2: Overview of DGPs used in simulation studies.

DGP	Distribution	Observation density	Moment	Link function
Count	Poisson	$\lambda_t \exp(-\lambda_t) / y_t!$	$\mathbb{E}y_t = \lambda_t$	$\lambda_t = \exp(\alpha_t)$
Count	Negative binomial	$\frac{\Gamma(k+y_t)}{\Gamma(k)\Gamma(y_t+k)} \left(\frac{k}{k+\lambda_t}\right)^k \left(\frac{\lambda_t}{k+\lambda_t}\right)^{y_t}$	$\mathbb{E}y_t = \lambda_t$	$\lambda_t = \exp(\alpha_t)$
Intensity	Exponential	$\lambda_t \exp(-\lambda_t y_t)$	$\mathbb{E}y_t = \lambda_t^{-1}$	$\lambda_t = \exp(\alpha_t)$
Duration	Gamma	$\{\Gamma(k)\beta_t^k\}^{-1} y_t^{k-1} \exp(-y_t/\beta_t)$	$\mathbb{E}y_t = k\beta_t$	$\beta_t = \exp(\alpha_t)$
Duration	Weibull	$k/\beta_t (y_t/\beta_t)^{k-1} \exp\{- (y_t/\beta_t)^k\}$	$\mathbb{E}y_t = \beta_t \Gamma(1 + 1/k)$	$\beta_t = \exp(\alpha_t)$
Volatility	Gaussian	$\{2\pi\sigma_t^2\}^{-1/2} \exp\{-y_t^2/(2\sigma_t^2)\}$	$\mathbb{E}y_t^2 = \sigma_t^2$	$\sigma_t^2 = \exp(\alpha_t)$
Volatility	Student's $t$	$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{(\nu-2)\pi}\Gamma(\nu/2)\sigma_t} \left(1 + \frac{y_t^2}{(\nu-2)\sigma_t^2}\right)^{-\frac{\nu+1}{2}}$	$\mathbb{E}y_t^2 = \sigma_t^2$	$\sigma_t^2 = \exp(\alpha_t)$
Dependence	Gaussian	$\frac{1}{2\pi\sqrt{1-\rho_t^2}} \exp\left\{-\frac{y_{1t}^2+y_{2t}^2-2\rho_t y_{1t}y_{2t}}{2(1-\rho_t^2)}\right\}$	$\mathbb{E}y_{1t}y_{2t} = \rho_t$	$\rho_t = \frac{1-\exp(-\alpha_t)}{1+\exp(-\alpha_t)}$
Dependence	Student's $t$	$\frac{\nu}{2\pi(\nu-2)\sqrt{1-\rho_t^2}} \left(1 + \frac{y_{1t}^2+y_{2t}^2-2\rho_t y_{1t}y_{2t}}{2(1-\rho_t^2)}\right)^{-\frac{\nu+2}{2}}$	$\mathbb{E}y_{1t}y_{2t} = \rho_t$	$\rho_t = \frac{1-\exp(-\alpha_t)}{1+\exp(-\alpha_t)}$
Local level	Student's $t$	$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{(\nu-2)\pi}\Gamma(\frac{\nu}{2})\sigma} \left(1 + \frac{(y_t-\mu_t)^2}{(\nu-2)\sigma^2}\right)^{-\frac{\nu+1}{2}}$	$\mathbb{E}y_t = \mu_t$	$\mu_t = \alpha_t$

Note: The table contains ten data-generating processes (DGPs) and link functions, the first nine of which are adapted from Koopman et al. (2016). For each model, the DGP is given by the linear Gaussian state equation (9) in combination with the observation density and link functions indicated in the table.

Table 3: Scores and information quantities for DGPs in Table 2.

DGP	Distribution	Score	Realised information	Information
Type	Distribution	$\frac{d\ell(y_t \alpha_t)}{d\alpha_t}$	$-\frac{d^2\ell(y_t \alpha_t)}{d\alpha_t^2}$	$\mathbb{E}\left[-\frac{d^2\ell(y_t \alpha_t)}{d\alpha_t^2}\middle \alpha_t\right]$
Count	Poisson	$\frac{y_t - \lambda_t}{y_t - \lambda_t}$	$\lambda_t$	$\lambda_t$
Count	Negative binomial	$\frac{y_t - \lambda_t(k + y_t)/(k + \lambda_t)}{1 - \lambda_t y_t}$	$k\lambda_t(k + y_t)/(k + \lambda_t)^2$	$k\lambda_t/(k + \lambda_t)$
Intensity	Exponential	$\frac{y_t/\beta_t - k}{k(y_t/\beta_t)^k - k}$	$y_t\lambda_t$	1
Duration	Gamma	$\frac{y_t^2/(2\sigma_t^2) - 1/2}{\omega_t y_t^2/(2\sigma_t^2) - 1/2}$	$\frac{y_t/\beta_t}{k^2(y_t/\beta_t)^k}$	$k$
Duration	Weibull	$\frac{\rho_t}{2} + \frac{1}{2} \frac{z_{1t} z_{2t}}{1 - \rho_t^2}$	$y_t^2/(2\sigma_t^2)$	$k^2$
Volatility	Gaussian	$\omega_t := (\nu + 1)/(\nu - 2 + y_t^2/\sigma_t^2)$	$(\nu - 2)(\nu + 1)/2 \frac{y_t^2/\sigma_t^2}{(\nu - 2 + y_t^2/\sigma_t^2)^2}$	1/2
Volatility	Student's $t$	$\frac{\rho_t}{2} + \frac{1}{2} \frac{z_{1t} z_{2t}}{1 - \rho_t^2}$	$0 \not\leq \frac{1}{4} \frac{z_{1t}^2 + z_{2t}^2}{1 - \rho_t^2} - \frac{1 - \rho_t^2}{4}$	$\nu/(2\nu + 6)$
Dependence	Gaussian	$z_{1t} := y_{1t} - \rho_t y_{2t}, \quad z_{2t} := y_{2t} - \rho_t y_{1t}$	$0 \not\leq \frac{1}{4} \frac{z_{1t}^2 + z_{2t}^2}{1 - \rho_t^2} - \frac{1 - \rho_t^2}{4}$	$(1 + \rho_t^2)/4$
Dependence	Student's $t$	$\frac{\rho_t}{2} + \frac{\omega_t}{2} \frac{z_{1t} z_{2t}}{1 - \rho_t^2}$	$0 \not\leq \frac{\omega_t}{4} \frac{z_{1t}^2 + z_{2t}^2}{1 - \rho_t^2} - \frac{1}{2} \frac{\omega_t^2}{\nu + 2} \frac{z_{1t}^2 z_{2t}^2}{(1 - \rho_t^2)^2} - \frac{1 - \rho_t^2}{4}$	$\frac{2 + \nu(1 + \rho_t^2)}{4(\nu + 4)}$
Local level	Student's $t$	$\frac{\nu + 1}{\sigma} \frac{\nu - 2 + (y_t - \mu_t)^2/\sigma^2}{\nu - 2 + \frac{y_{1t}^2 + y_{2t}^2 - 2\rho_t y_{1t} y_{2t}}{2(1 - \rho_t^2)}}$	$z_{1t} := y_{1t} - \rho_t y_{2t}, \quad z_{2t} := y_{2t} - \rho_t y_{1t}$	$\frac{\nu(\nu + 1)}{\sigma^2(\nu - 2)(\nu + 3)}$

Note: The table displays the scores, realised information quantities and expected information quantities for the data-generating processes in Table 2. The realised information quantities are positive semi-definite, except for the bottom three, as indicated by  $0 \not\leq \dots$ . We deviate from Koopman et al. (2016) in computing scores and information quantities with respect to the state  $\alpha_t$ , which is subject to linear Gaussian dynamics, rather than with respect to its transformation given by  $\lambda_t, \beta_t, \sigma_t^2$  or  $\rho_t$ .

For each DGP, we simulate 1,000 time series of length 5,000, where the true constant (hyper)parameter  $\psi$  can be read off from Table 5, which also contains our average parameter estimates across 1,000 simulations. For the first nine DGPs, the true parameters are taken from Koopman et al. (2016), who argue that these parameters are representative of those found in empirical work.

Following Koopman et al. (2016), we use the first 2,500 observations of each time series to estimate the parameters. We compute the approximate MLE (26), which typically takes no longer than one second using a gradient-based numerical optimiser. Using these parameter estimates, we then run the Bellman filter on the entire data set, including the last 2,500 observations, which typically takes a mere fraction of a second. Finally, we evaluate mean absolute errors (MAEs) of the last 2,500 filtered states compared to the last 2,500 true (simulated) states. By doing so, we ensure that our filtered states can genuinely be viewed as ‘out-of-sample’ estimates. In total,  $2,500 \times 1,000$  simulations = 2.5 million such out-of-sample filtered states are computed for each DGP.

To judge the performance of the Bellman filter, we also report the MAE achieved by the optimal estimator (4) evaluated at the true parameters. Solving  $t$  first-order conditions for *each* time  $t$  between  $t = 2,501$  and  $t = 5,000$  turns out to be excessively time consuming. Instead of using an expanding window, therefore, we compute the optimal filter (4) for each time  $t$  using a moving window of 250 observations, such that 250 first-order conditions are solved for each time  $t$ , resulting in 250 state estimates, of which only the final state estimate is used in the computation of the MAE. The performance of the mode estimator does not noticeably deteriorate as a result. With this change, finding the posterior mode of length 250 for all times  $t$  between  $t = 2,501$  and  $t = 5,000$  takes around 30 seconds for any individual time series, which we deem acceptable. Even then, evaluating the posterior mode for all 1,000 samples from any DGP takes around  $1,000 \times 30$  seconds  $/60^2 \approx 8$  hours. For both dependence models, the total computing time exceeds 24 hours. Parameter estimation is infeasible using the optimal filter (4).

Table 4 reports MAEs resulting from the (a) optimal filter (4) evaluated at the true parameters, (b) Bellman filter in Table 1 using three different estimation settings, and, for some models, (c) quasi maximum likelihood estimation (QMLE) of the Kalman filter. MAEs are reported relative to those of the optimal estimator (4). Evaluated at the true parameters, the MAEs of Bellman-filtered states resemble those of the optimal state estimates, being at most  $\sim 1\%$  higher for all DGPs considered. When the parameters are estimated in an in-sample setting, the MAEs of the Bellman-filtered states are at most  $\sim 1.3\%$  higher. Finally, even when the parameters are estimated in an out-of-sample setting, the MAEs of the Bellman-filtered states remain within 2% of that of the optimal state estimates for eight out of 10 DGPs, while never exceeding the MAE of the optimal estimator by more than  $\sim 3.4\%$ .

In comparing the performance of the Bellman filter in three settings — evaluated at the true parameters, in-sample estimated parameters, and out-of-sample estimated parameters — the general conclusion is that only a small fraction of the additional MAE compared to the optimal method is caused by approximate filtering and estimation. Rather, most of the additional MAE is caused by the design choice that the parameter estimation uses only the first half of the data, whereas the evaluation of MAEs pertains to the second half. No feasible method can eliminate this source of error. Given that the error originating from approximate filtering and approximate estimation is extremely limited — typically less than one percent — we interpret Table 4 as strong evidence in

Table 4: Mean absolute errors (MAEs) of filtered states in simulation studies.

DGP		Optimal but infeasible estimator (4)		Bellman filter			Kalman filter by QMLE
Type	Distribution	True parameters	True parameters	Estimated parameters (in-sample)	Estimated parameters (out-of-sample)	Estimated parameters (out-of-sample)	Estimated parameters (out-of-sample)
		Relative MAE		Relative MAE			Relative MAE
Count	Poisson	0.2827	1.0000	1.0043	0.9944	1.0014	n/a
Count	Neg binomial	0.2999	1.0000	1.0020	0.9984	1.0061	n/a
Intensity	Exponential	0.2864	1.0000	1.0050	1.0066	1.0148	n/a
Duration	Gamma	0.2590	1.0000	1.0041	1.0066	1.0121	n/a
Duration	Weibull	0.2638	1.0000	1.0048	0.9949	1.0005	n/a
Volatility	Gaussian	0.3374	1.0000	1.0044	0.9945	1.0048	1.1903
Volatility	Student's $t$	0.3521	1.0000	1.0021	1.0017	1.0135	1.1882
Dependence	Gaussian	0.2955	1.0000	1.0028	1.0132	1.0342	n/a
Dependence	Student's $t$	0.3017	1.0000	1.0017	1.0074	1.0303	n/a
Local level	Student's $t$	0.1591	1.0000	1.0097	1.0079	1.0110	1.1118

*Note:* We simulated 1,000 time series each of length 5,000 for 10 data-generating processes of type (9) (the observation densities are listed in Table 2). To estimate the constant (hyper)parameters, we used estimator (26) based on the first 2,500 observations (out-of-sample estimation) or the last 2,500 observations (in-sample estimation). Mean absolute errors (MAEs) were computed for the last 2,500 states by comparing optimal estimates  $\tilde{a}_{t|t}$  obtained from equation (4) and the Bellman-filtered states  $a_{t|t}$  versus the true (simulated) states  $\alpha_t$ . For the optimal estimator (4), we used the most recent 250 observations to solve a system of 250 first-order conditions and extracted the final state estimate  $\tilde{a}_{t|t}$  as the optimal state estimate for each time  $t$ . For three models, we report the MAE obtained by quasi maximum likelihood estimation (QMLE) of the Kalman filter; the last column shows 'n/a' if, to the best of our knowledge, no alternative computationally efficient approximate filter is available.

Table 5: Average parameter estimates in simulation studies.

DGP		True parameters versus average parameter estimates								
Type	Distribution	True parameters								
		$c$	$T$	$Q$	$Q$	$\nu^{-1}$	$\nu^{-1}$	$k$	$k$	$\sigma$
		0.0000	0.9800	0.0225	0.0100	0.1000	0.3333	1.2000	4.0000	0.4500
Count	Poisson	-0.0073	0.9765	0.0236	n/a	n/a	n/a	n/a	n/a	n/a
Count	Negative binomial	-0.0060	0.9772	0.0238	n/a	n/a	n/a	n/a	4.2668	n/a
Intensity	Exponential	0.0015	0.9796	0.0197	n/a	n/a	n/a	n/a	n/a	n/a
Duration	Gamma	-0.0010	0.9791	0.0205	n/a	n/a	n/a	1.4831	n/a	n/a
Duration	Weibull	0.0089	0.9758	0.0256	n/a	n/a	n/a	1.2071	n/a	n/a
Volatility	Gaussian	0.0069	0.9745	0.0280	n/a	n/a	n/a	n/a	n/a	n/a
Volatility	Student's $t$	0.0044	0.9764	0.0252	n/a	0.0865	n/a	n/a	n/a	n/a
Dependence	Gaussian	0.0000	0.9818	n/a	0.0065	n/a	n/a	n/a	n/a	n/a
Dependence	Student's $t$	-0.0001	0.9799	n/a	0.0077	0.0943	n/a	n/a	n/a	n/a
Local level	Student's $t$	0.0001	0.9741	0.0264	n/a	n/a	0.3193	n/a	n/a	0.4924

*Note:* Reported are the average parameter estimates across 1,000 simulations based on the approximate maximum likelihood estimator (26). The data-generating processes are defined in Table 2. For the simulation setting, see the note below Table 4. For each DGP, only a subset of all parameters is relevant; “n/a” is indicated for parameters that do not apply.

support of the Bellman filter in Table 1 as well as the approximate MLE (26).

It is no coincidence that the optimal filtered estimates  $\tilde{a}_{t|t}$  and the Bellman-filtered states  $a_{t|t}$  produce approximately the same MAEs compared to the true states  $\alpha_t$ . The Bellman-filtered states  $a_{t|t}$  are constructed for the express purpose of imitating the optimal state estimates  $\tilde{a}_{t|t}$ . Indeed, the correlation between the optimal state estimates  $\tilde{a}_{t|t}$  and the Bellman-filtered states  $a_{t|t}$  usually exceeds 99%, such that these are virtually identical (this number is not shown in the table).

Focusing on both stochastic volatility models, the MAE of the Bellman-filtered states remains within  $\sim 1.4\%$  of that of the optimal estimator, irrespective of the estimation setting. As an alternative, computationally efficient estimator, we consider the common practice of squaring the observations and taking logarithms, leading to a linear state-space model, albeit with biased and non-Gaussian observation noise (for details, see Ruiz, 1994 or Harvey et al., 1994). The advantage is that filtered state estimates can be obtained via QMLE of the Kalman filter. However, as the last column of Table 4 shows, the resulting MAEs are substantially worse than those produced by the Bellman filter, lagging  $\sim 19\%$  behind the optimal estimator. It could be argued that the choice of MAE loss functions causes the Kalman filter to underperform compared to mode-based estimators. However, even when reporting root mean squared errors (RMSEs), the results for the Kalman filter are hardly more favourable (see Appendix F). Conversely, the performance of the Bellman-filtered states remains within  $\sim 1.4\%$  of that of estimator (4) measured in terms of either MAE or RMSE.

Focusing on the local-level model, the performance of the Bellman filter is within  $\sim 1\%$  of that of the optimal estimator, even at parameters estimated out-of-sample. In contrast, the Kalman filter, sent astray by heavy-tailed observation noise, lags  $\sim 11\%$  behind the optimal estimator. Again, this difference is not due to the choice of loss function; the relative performance of the Kalman filter deteriorates further if we report RMSEs (see Appendix F). The  $\sim 10\%$  difference in performance between the Bellman and Kalman filters occurs in spite of the fact that both filters are, most of the time, in close agreement. However, the *maximum* absolute error in the out-of-sample period,

averaged across 1,000 samples, is 1.75 for the Kalman filter; more than twice that for the Bellman filter (0.77). This shows that the Bellman filter is more robust in the face of heavy-tailed observation noise, while having only a single additional parameter to estimate (the degrees of freedom of the observation noise,  $\nu$ ). While several filters have been constructed in an ad hoc manner for the express purpose of being robust (e.g. Harvey and Luati, 2014 and Calvet et al., 2015), in our case robustness follows automatically from Bellman’s equation (14) along with the fact that the score  $d\ell(y_t|a_t)/da_t$  for a Student’s  $t$  distribution, where  $a_t$  represents the location, is bounded.

For three DGPs, we were able compare the performance of the Bellman filter with an in terms of computation equally efficient method: QMLE of the Kalman filter. To the best of our knowledge, no computationally efficient filters are available for the remaining seven DGPs. For these models, the Bellman filter has a unique advantage compared to other available methods. While computationally intensive methods may outperform the Bellman filter, they cannot do so by much, as the minimal gap between the performance of the Bellman filter and that of the (theoretically optimal) mode estimator leaves little room to exploit. In finite samples, even the most computationally intensive methods will not be able to close this gap entirely. Depending on the performance required, the additional effort of setting up and executing more computationally taxing methods may or may not be worthwhile. If deemed worthwhile, the parameter estimates and state estimates obtained from the Bellman filter may provide a valuable starting point.

Table 5 reports the average parameter estimates across 1,000 simulations of each DGP obtained from the approximate MLE (26). The average parameter estimates are reasonably close to the true values. Given that these parameter estimates can typically be obtained very quickly using a gradient-based numerical optimiser, they may prove useful in their own right but also as starting points for computationally more intensive methods.

Finally, Appendix G demonstrates that approximate confidence intervals implied by the Bellman filter, i.e. with endpoints given by  $a_{t|t} \pm 2/\sqrt{I_{t|t}}$  for each time step, tend to be fairly accurate, containing the true states 90 – 95% of the time across all 10 DGPs.

## 7 Conclusion

The Bellman filter for state-space models as developed in this article generalises the Kalman filter and is equally computationally inexpensive, but is robust in the case of heavy-tailed observation noise and applicable to a wider range of dynamic models involving e.g. count, intensity, duration, volatility and dependence. Unlike the posterior mode, from which it is derived, the Bellman filter can be applied in real time and remains feasible in practice when (hyper)parameters must be estimated.

We investigated the performance of the Bellman filter in extensive simulation studies involving a wide range of data-generating processes. Broadly speaking, we sacrifice little in terms of optimality when compared to the (theoretically) optimal — but generally infeasible — mode estimator, which is evaluated at the true parameters. The mean absolute errors (MAEs) of Bellman-filtered states are only marginally worse — at most  $\sim 3\%$  — from those produced by the optimal estimator. This remains true even for parameters estimated in a training set distinct from the evaluation set.

Finally, we note that making different choices with respect to (a) the method used to (parametrically) approximate value functions and (b) the optimisation routine used to find the argmax would

result in different Bellman filters. In general, therefore, the Bellman filter can be indexed through the chosen approximating function and chosen optimisation routine. This article has explored only a small part of this space: we have found it sufficiently accurate to approximate value functions using multivariate quadratic functions, and to make use of plain-vanilla optimisation schemes. There may be situations in which a more sophisticated approach is warranted; something we intend to explore in the future.

## A Proof of Proposition 1

Standard dynamic programming arguments imply

$$\begin{aligned}
V_t(\mathbf{a}_t) &:= \max_{\mathbf{a}_{1:t-1} \in \mathbb{R}^{m \times (t-1)}} \ell_t(\mathbf{a}_{1:t}), \text{ by definition (6),} \\
&= \max_{\mathbf{a}_{1:t-1} \in \mathbb{R}^{m \times (t-1)}} \{ \ell(\mathbf{y}_t | \mathbf{a}_t) + \ell(\mathbf{a}_t | \mathbf{a}_{t-1}) + \ell_{t-1}(\mathbf{a}_{1:t-1}) \}, \text{ by recursion (5),} \\
&= \max_{\mathbf{a}_{t-1} \in \mathbb{R}^m} \{ \ell(\mathbf{y}_t | \mathbf{a}_t) + \ell(\mathbf{a}_t | \mathbf{a}_{t-1}) + \max_{\mathbf{a}_{1:t-2} \in \mathbb{R}^{m \times (t-2)}} \ell_{t-1}(\mathbf{a}_{1:t-1}) \}, \\
&\quad \text{by moving all but one maximisation inside curly brackets,} \\
&= \max_{\mathbf{a}_{t-1} \in \mathbb{R}^m} \{ \ell(\mathbf{y}_t | \mathbf{a}_t) + \ell(\mathbf{a}_t | \mathbf{a}_{t-1}) + V_{t-1}(\mathbf{a}_{t-1}) \}, \text{ again by definition (6),} \\
&= \ell(\mathbf{y}_t | \mathbf{a}_t) + \max_{\mathbf{a}_{t-1} \in \mathbb{R}^m} \{ \ell(\mathbf{a}_t | \mathbf{a}_{t-1}) + V_{t-1}(\mathbf{a}_{t-1}) \}.
\end{aligned}$$

Further, it is evident that

$$\mathbf{a}_{t|t} := \arg \max_{\mathbf{a}_t \in \mathbb{R}^m} V_t(\mathbf{a}_t) = \arg \max_{\mathbf{a}_t \in \mathbb{R}^m} \max_{\mathbf{a}_{1:t-1} \in \mathbb{R}^{m \times (t-1)}} \ell_t(\mathbf{a}_{1:t}) = \tilde{\mathbf{a}}_{t|t}, \quad (27)$$

where  $\tilde{\mathbf{a}}_{t|t}$  was defined in equation (3).

## B Derivation of equation (14)

In principle, equation (14) can be obtained by substituting equation (13) into equation (12) and performing algebraic manipulations. The desired result can be obtained more elegantly by ‘completing the square’ as follows. First, we replace  $\mathbf{a}_t$  by  $\mathbf{a}_t^*$  in equation (12), which then contains the following terms:

$$-\frac{1}{2}(\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1}^*)' \mathbf{Q}^{-1} (\mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1}^*) - \frac{1}{2}(\mathbf{a}_{t-1}^* - \mathbf{a}_{t-1|t-1})' \mathbf{I}_{t-1|t-1} (\mathbf{a}_{t-1}^* - \mathbf{a}_{t-1|t-1}). \quad (28)$$

Then we recall from equation (13) that  $\mathbf{a}_{t-1}^*$  is linear in  $\mathbf{a}_t$ , such that the collection of terms in equation (28) above is at most multivariate quadratic in  $\mathbf{a}_t$ . Hence, we should be able to re-write equation (28) as a quadratic function (i.e., by ‘completing the square’) as follows:

$$-\frac{1}{2}(\mathbf{a}_t - \mathbf{a}_{t|t-1})' \mathbf{I}_{t|t-1} (\mathbf{a}_t - \mathbf{a}_{t|t-1}) + \text{constants}, \quad (29)$$

for some vector  $\mathbf{a}_{t|t-1}$  to be found and some matrix  $\mathbf{I}_{t|t-1}$  to be determined.

To do this, we note that  $\mathbf{a}_{t|t-1}$  represents the argmax of equation (29), which can be found most easily by differentiating equation (28) with respect to  $\mathbf{a}_t$  and setting the result to zero. By the



envelope theorem, we need not account for the fact that  $\mathbf{a}_{t-1}^*$  depends on  $\mathbf{a}_t$  (the first derivative with respect to  $\mathbf{a}_{t-1}^*$  is zero because  $\mathbf{a}_{t-1}^*$  is optimal). Thus we set the derivative of equation (28) with respect to  $\mathbf{a}_t$  equal to zero, which gives  $\mathbf{0} = \mathbf{a}_t - \mathbf{c} - \mathbf{T}\mathbf{a}_{t-1}^*$ , or, by substituting  $\mathbf{a}_{t-1}^*$  from equation (13), we obtain

$$\begin{aligned} \mathbf{0} = \mathbf{a}_t - \mathbf{c} - \mathbf{T}[\mathbf{I}_{t-1|t-1} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]^{-1}\mathbf{I}_{t-1|t-1}\mathbf{a}_{t-1|t-1} \\ - \mathbf{T}[\mathbf{I}_{t-1|t-1} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]^{-1}\mathbf{T}'\mathbf{Q}^{-1}(\mathbf{a}_t - \mathbf{c}), \end{aligned} \quad (30)$$

It is easily verified that its solution is  $\mathbf{a}_{t|t-1} := \mathbf{T}\mathbf{a}_{t-1|t-1} + \mathbf{c}$ , which confirms prediction step (15).

Next, we compute the negative second derivative of equation (28) with respect to  $\mathbf{a}_t$ , which should give us  $\mathbf{I}_{t|t-1}$ . To account for the dependence of  $\mathbf{a}_{t-1}^*$  on  $\mathbf{a}_t$ , we use the chain rule. Specifically, equation (13) gives that  $\mathbf{a}_{t-1}^*$  is linear in  $\mathbf{a}_t$  with the following Jacobian matrix:

$$\mathbf{J} := \frac{d\mathbf{a}_{t-1}^*}{d\mathbf{a}_t'} = [\mathbf{I}_{t|t} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]^{-1}\mathbf{T}'\mathbf{Q}^{-1}. \quad (31)$$

Next, the chain rule tells us that

$$\frac{d^2 \cdot}{d\mathbf{a}_t d\mathbf{a}_t'} = \begin{bmatrix} \mathbf{1} \\ \mathbf{J} \end{bmatrix}' \begin{bmatrix} \frac{\partial^2 \cdot}{\partial \mathbf{a}_t \partial \mathbf{a}_t'} & \frac{\partial^2 \cdot}{d\mathbf{a}_t d\mathbf{a}_{t-1}^*'} \\ \frac{\partial^2 \cdot}{\partial \mathbf{a}_{t-1}^* \partial \mathbf{a}_t'} & \frac{\partial^2 \cdot}{\partial \mathbf{a}_{t-1}^* \partial \mathbf{a}_{t-1}^*'} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{J} \end{bmatrix}, \quad (32)$$

where instances of  $\partial$  and  $d$  denote ‘partial’ and ‘total’ derivatives, respectively, while  $\mathbf{1}$  denotes an identity matrix of appropriate size. As before, the envelope theorem ensures that no *first* derivative with respect to  $\mathbf{a}_t^*$  appears. When equation (32) is applied, we find that the negative second derivative of equation (28) becomes

$$\begin{aligned} & \begin{bmatrix} \mathbf{1} \\ \mathbf{J} \end{bmatrix}' \begin{bmatrix} \mathbf{Q}^{-1} & -\mathbf{Q}^{-1}\mathbf{T} \\ -\mathbf{T}'\mathbf{Q}^{-1} & \mathbf{I}_{t|t} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{J} \end{bmatrix} \\ &= \mathbf{Q}^{-1} - \underbrace{\mathbf{Q}^{-1}\mathbf{T}\mathbf{J}} - \underbrace{\mathbf{J}'\mathbf{T}'\mathbf{Q}^{-1}} + \underbrace{\mathbf{J}'[\mathbf{I}_{t|t} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]\mathbf{J}}, \\ &= \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{T}[\mathbf{I}_{t|t} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]^{-1}\mathbf{T}'\mathbf{Q}^{-1}. \end{aligned} \quad (33)$$

In the last line, we have used the fact that all three terms with curly brackets equal  $\mathbf{Q}^{-1}\mathbf{T}[\mathbf{I}_{t|t} + \mathbf{T}'\mathbf{Q}^{-1}\mathbf{T}]^{-1}\mathbf{T}'\mathbf{Q}^{-1}$ , such that two terms with curly brackets and opposite signs cancel, leaving only one term with a negative sign, which confirms prediction step (16).

## C Choice of optimisation method in Table 1

Newton’s method discussed in the main text is optimal if  $\mathbf{I}_{t|t-1} - d^2\ell(\mathbf{y}_t|\mathbf{a}_t)/(d\mathbf{a}_t d\mathbf{a}_t')$  is positive definite (p.d.), in which case its inverse determines the search direction and the step length. Assuming  $\mathbf{I}_{t|t-1}$  is constructed to be p.d., it is sufficient that the realised marginal information  $-d^2\ell(\mathbf{y}_t|\mathbf{a}_t)/(d\mathbf{a}_t d\mathbf{a}_t')$  is positive semi-definite (p.s.d.) for all observations  $\mathbf{y}_t$  and all states  $\mathbf{a}_t \in \mathbb{R}^m$ . If the realised marginal information fails to be p.s.d. for some values of  $\mathbf{y}_t$  and  $\mathbf{a}_t$ , then the difference  $\mathbf{I}_{t|t-1} - d^2\ell(\mathbf{y}_t|\mathbf{a}_t)/(d\mathbf{a}_t d\mathbf{a}_t')$  may (but need not) fail to be p.d. In such cases, to guarantee well-defined optimisation steps, we may consider Fisher’s scoring method in Table 1, which is applicable

more generally than Newton’s method because the expected (as opposed to realised) information is always p.s.d. if the model is identified.

As can be seen from Table 1, the convergence of either Newton’s or Fisher’s method implies that the (same) first-order condition is satisfied at the argmax  $\mathbf{a}_{t|t}$ , namely  $d\ell(\mathbf{y}_t|\mathbf{a}_{t|t})/d\mathbf{a}_{t|t} = \mathbf{I}_{t|t-1}(\mathbf{a}_{t|t} - \mathbf{a}_{t|t-1})$ . Hence both routines may converge to the same (possibly local) optimum. For both routines, the matrix of second derivatives evaluated at the optimum is expected to be negative definite. If we iterate until convergence, therefore, we should be able to use Newton’s method to update the information matrix, as indicated under the method ‘Newton’ under the step ‘Update’. For some models, however, involving e.g. time-varying correlation, the realised marginal information is unbounded above (see Table 3). In such cases, it is advisable to use the more conservative updating step that corresponds to Fisher’s method, as indicated under the step ‘Update’ in Table 1 in the main text.

Finally, the Berndt-Hall-Hall-Hausman (BHHH) algorithm in Table 1 may be useful if second derivatives are hard to derive, or if the state dimension is large; the BHHH updating step is attractive because the required inverse can be computed in closed form using a lemma by Sherman and Morrison (1950). Additionally, the BHHH algorithm may be attractive if the score is unbounded, in which case the BHHH updating step still ensures step sizes of moderate length, such that the optimisation does not stray too far from its starting point.

In all cases, convergence would be guaranteed if some regularity conditions were fulfilled, and if the algorithms were supplemented by a line search (e.g. Nocedal and Wright, 2006). In practical applications, we have found this unnecessary; for practical purposes, even performing a fixed number of Newton or Fisher optimisation steps — five, say — typically suffices.

## D Kalman filter as a special case

To see that the Kalman filter is a special case of the Bellman filter in Table 1, take  $\mathbf{y}_t = \mathbf{d} + \mathbf{Z}\mathbf{a}_t + \boldsymbol{\varepsilon}_t$  with  $\boldsymbol{\varepsilon}_t \sim \text{i.i.d. } \mathbf{N}(\mathbf{0}, \mathbf{H})$ . Then

$$\ell(\mathbf{y}_t|\mathbf{a}_t) = -1/2(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}\mathbf{a}_t)' \mathbf{H}^{-1}(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}\mathbf{a}_t) + \text{constants}. \quad (34)$$

The score and realised marginal information are

$$\frac{d\ell(\mathbf{y}_t|\mathbf{a}_t)}{d\mathbf{a}_t} = \mathbf{Z}' \mathbf{H}^{-1}(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}\mathbf{a}_t), \quad -\frac{d^2\ell(\mathbf{y}_t|\mathbf{a}_t)}{d\mathbf{a}_t d\mathbf{a}_t'} = \mathbf{Z}' \mathbf{H}^{-1} \mathbf{Z}. \quad (35)$$

As the realised information is constant, it equals the (expected) marginal information. Taking the starting point  $\mathbf{a}_{t|t}^{(0)} = \mathbf{a}_{t|t-1}$  for Newton’s optimisation method, the estimate after a single Newton iteration reads

$$\mathbf{a}_{t|t}^{(1)} = \mathbf{a}_{t|t-1} + (\mathbf{I}_{t|t-1} + \mathbf{Z}' \mathbf{H}^{-1} \mathbf{Z})^{-1} \mathbf{Z}' \mathbf{H}^{-1}(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}\mathbf{a}_{t|t-1}), \quad (36)$$

which is exactly the Kalman filter level update written in information form. To see the equivalence with the covariance form of the Kalman filter, suppose that  $\mathbf{P}_{t|t-1} := \mathbf{I}_{t|t-1}^{-1}$  exists. Then, using the Woodbury matrix inversion formula, see e.g. Henderson and Searle (1981, eq. 1), the expression above is equivalent to

$$\mathbf{a}_{t|t}^{(1)} = \mathbf{a}_{t|t-1} + \mathbf{P}_{t|t-1} \mathbf{Z}' (\mathbf{Z} \mathbf{P}_{t|t-1} \mathbf{Z}' + \mathbf{H})^{-1} (\mathbf{y}_t - \mathbf{d} - \mathbf{Z}\mathbf{a}_{t|t-1}), \quad (37)$$

which is exactly the Kalman filter updating step, see e.g. Harvey (1990, p. 106). For the information matrix update we have

$$\mathbf{I}_{t|t} = \mathbf{I}_{t|t-1} - \left. \frac{d^2 \ell(\mathbf{y}_t | \mathbf{a})}{d\mathbf{a} d\mathbf{a}'} \right|_{\mathbf{a}=\mathbf{a}_{t|t}} = \mathbf{I}_{t|t-1} + \mathbf{Z}' \mathbf{H}^{-1} \mathbf{Z}. \quad (38)$$

If the inverses  $\mathbf{P}_{t|t-1} := \mathbf{I}_{t|t-1}^{-1}$  and  $\mathbf{P}_{t|t} := \mathbf{I}_{t|t}^{-1}$  exist, then, using again Henderson and Searle (1981, eq. 1), we find

$$\mathbf{P}_{t|t} = \mathbf{I}_{t|t}^{-1} = (\mathbf{I}_{t|t-1} + \mathbf{Z}' \mathbf{H}^{-1} \mathbf{Z})^{-1} = \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1} \mathbf{Z}' (\mathbf{Z} \mathbf{P}_{t|t-1} \mathbf{Z}' + \mathbf{H})^{-1} \mathbf{Z} \mathbf{P}_{t|t-1}, \quad (39)$$

which is exactly the Kalman filter covariance matrix updating step, see again Harvey (1990, p. 106).

## E Iterated extended Kalman filter as a special case

To see that the iterated extended Kalman filter is a special case of the Bellman filter in Table 1, take  $\mathbf{y}_t = \mathbf{d} + \mathbf{Z}(\boldsymbol{\alpha}_t) + \boldsymbol{\varepsilon}_t$  with  $\boldsymbol{\varepsilon}_t \sim$  i.i.d.  $N(\mathbf{0}, \mathbf{H})$ . Here,  $\mathbf{Z}_t := \mathbf{Z}(\boldsymbol{\alpha}_t)$  is column vector of the same size as  $\mathbf{y}_t$ , where each element of  $\mathbf{Z}_t$  depends on the elements of  $\boldsymbol{\alpha}_t$ . Then

$$\ell(\mathbf{y}_t | \mathbf{a}_t) = -1/2(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}(\mathbf{a}_t))' \mathbf{H}^{-1}(\mathbf{y}_t - \mathbf{d} - \mathbf{Z}(\mathbf{a}_t)) + \text{constants}. \quad (40)$$

The score and marginal information are similar to those in Appendix ??, as long as  $\mathbf{Z}$  there is replaced by the Jacobian of the transformation from  $\boldsymbol{\alpha}_t$  to  $\mathbf{Z}_t$ , that is  $d\mathbf{Z}(\mathbf{a}_t)/d\mathbf{a}'_t$ . Hence

$$\frac{d \ell(\mathbf{y}_t | \mathbf{a}_t)}{d\mathbf{a}_t} = \frac{d\mathbf{Z}'}{d\mathbf{a}_t} \mathbf{H}^{-1} (\mathbf{y}_t - \mathbf{d} - \mathbf{Z}(\mathbf{a}_t)), \quad (41)$$

$$\frac{d^2 \ell(\mathbf{y}_t | \mathbf{a}_t)}{d\mathbf{a}_t d\mathbf{a}'_t} = -\frac{d\mathbf{Z}'}{d\mathbf{a}_t} \mathbf{H}^{-1} \frac{d\mathbf{Z}}{d\mathbf{a}'_t} + \text{second-order derivatives} \quad (42)$$

The iterated extended Kalman filter (IEKF) is obtained from the Bellman filter by choosing Newton's method and by making one further simplifying approximation: namely that all second-order derivatives of elements of  $\mathbf{Z}_t$  with respect to the elements of  $\boldsymbol{\alpha}_t$  are zero. It is not obvious under what circumstances this approximation is justified, but here we are interested only in showing that the IEKF is a special case of the Bellman filter. Hence, the IEKF is obtained by plugging the score and the realised marginal information matrix into the Bellman filter in Table 1 with Newton's iteration method, and taking the starting point  $\mathbf{a}_t^{(0)} = \mathbf{a}_{t|t-1}$ . Higher-order IEKFs may be obtained by retaining the second-order derivatives.

## F Root mean squared errors in simulation studies

Table 6: Root mean squared errors (RMSEs) of filtered states in simulation studies.

DGP		Estimator (4) in main text		Bellman filter			Kalman filter by QMLE
		True parameters	True parameters	Estimated parameters (in-sample)	Estimated parameters (out-of-sample)	Estimated parameters (out-of-sample)	
Type	Distribution	RMSE	Relative RMSE	Relative RMSE			Relative RMSE
Count	Poisson	0.3554	1.0000	1.0004	1.0001	1.0045	n/a
Count	Neg binomial	0.3765	1.0000	1.0007	1.0007	1.0056	n/a
Intensity	Exponential	0.3572	1.0000	1.0029	1.0042	1.0096	n/a
Duration	Gamma	0.3232	1.0000	1.0019	1.0029	1.0069	n/a
Duration	Weibull	0.3290	1.0000	1.0004	1.0010	1.0042	n/a
Volatility	Gaussian	0.4197	1.0000	1.0003	1.0010	1.0058	1.1847
Volatility	Student's $t$	0.4387	1.0000	1.0003	1.0037	1.0102	1.1416
Dependence	Gaussian	0.3672	1.0000	1.0024	1.0133	1.0244	n/a
Dependence	Student's $t$	0.3737	1.0000	1.0013	1.0071	1.0185	n/a
Local level	Student's $t$	0.2035	1.0000	1.0093	1.0069	1.0096	1.1482

*Note:* See the note under Table 4 in the main text. The only difference is that here we report root mean squared errors (RMSEs), not mean absolute errors (MAEs).

## G Coverage in simulation studies

Table 7: Coverage of Bellman-filter implied confidence intervals in simulation studies.

DGP		Bellman filter
		Estimated parameters
		Out-of-sample
Type	Distribution	Coverage of confidence interval (%)
Count	Poisson	95.3
Count	Negative binomial	95.2
Intensity	Exponential	93.5
Duration	Gamma	94.1
Duration	Weibull	93.2
Volatility	Gaussian	96.0
Volatility	Student's $t$	94.1
Dependence	Gaussian	89.6
Dependence	Student's $t$	91.5
Local level	Student's $t$	95.6

*Note:* See the note under Table 4 in the main text. This table reports how often the true states  $\alpha_t$  were found to be inside the approximate confidence interval for  $t = 2,501, \dots, 5000$ , where the approximate confidence intervals were derived from the Bellman filter in Table 1 of the main text with estimated parameters based on the first 2,500 observations (i.e., out-of-sample parameter estimation) using estimator (26) in the main text. For scalar states, the endpoints of the approximate confidence interval are given by  $a_{t|t} \pm 2/\sqrt{I_{t|t}}$ .

## References

- Anderson, B. D. and Moore, J. B. (2012) *Optimal Filtering*. Courier Corporation.
- Andrieu, C., Doucet, A. and Holenstein, R. (2010) Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**, 269–342.
- Bauwens, L. and Hautsch, N. (2006) Stochastic conditional intensity processes. *Journal of Financial Econometrics*, **4**, 450–493.
- Bauwens, L. and Veredas, D. (2004) The stochastic conditional duration model: A latent variable model for the analysis of financial durations. *Journal of Econometrics*, **119**, 381–412.
- Bellman, R. E. (1957) *Dynamic Programming*. Courier Dover Publications.
- Bertsekas, D. P. (2012) *Dynamic Programming and Optimal Control, Vol. II: Approximate Dynamic Programming*. Athena Scientific.
- Bunch, P. and Godsill, S. (2016) Approximations of the optimal importance density using Gaussian particle flow importance sampling. *Journal of the American Statistical Association*, **111**, 748–762.
- Calvet, L. E., Czellar, V. and Ronchetti, E. (2015) Robust filtering. *Journal of the American Statistical Association*, **110**, 1591–1606.
- Chen, R. and Liu, J. S. (2000) Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **62**, 493–508.
- De Valpine, P. (2004) Monte Carlo state-space likelihoods by weighted posterior kernel density estimation. *Journal of the American Statistical Association*, **99**, 523–536.
- Durbin, J. (1997) Optimal estimating equations for state vectors in non-Gaussian and nonlinear state space time series models. *Lecture Notes-Monograph Series*, 285–291.
- Durbin, J. and Koopman, S. J. (1997) Monte Carlo maximum likelihood estimation for non-Gaussian state space models. *Biometrika*, **84**, 669–684.
- (2000) Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **62**, 3–56.
- (2002) A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, **89**, 603–616.
- (2012) *Time Series Analysis by State Space Methods*. Oxford University Press.
- Fahrmeir, L. (1992) Posterior mode estimation by extended Kalman filtering for multivariate dynamic generalized linear models. *Journal of the American Statistical Association*, **87**, 501–509.
- Fahrmeir, L. and Kaufmann, H. (1991) On Kalman filtering, posterior mode estimation and Fisher scoring in dynamic exponential family regression. *Metrika*, **38**, 37–60.

- Fearnhead, P. and Clifford, P. (2003) On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **65**, 887–899.
- Frühwirth-Schnatter, S. and Wagner, H. (2006) Auxiliary mixture sampling for parameter-driven models of time series of counts with applications to state space modelling. *Biometrika*, **93**, 827–841.
- Ghysels, E., Harvey, A. C. and Renault, E. (1996) Stochastic volatility. In *Handbook of Statistics*, vol. 14, 119–191. Elsevier.
- Godsill, S., Doucet, A. and West, M. (2001) Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics*, **53**, 82–96.
- Godsill, S. J., Doucet, A. and West, M. (2004) Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, **99**, 156–168.
- Guarniero, P., Johansen, A. M. and Lee, A. (2017) The iterated auxiliary particle filter. *Journal of the American Statistical Association*, **112**, 1636–1647.
- Hafner, C. M. and Manner, H. (2012) Dynamic stochastic copula models: Estimation, inference and applications. *Journal of Applied Econometrics*, **27**, 269–295.
- Hamilton, J. D. (1989) A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, 357–384.
- Harvey, A. and Luati, A. (2014) Filtering with heavy tails. *Journal of the American Statistical Association*, **109**, 1112–1122.
- Harvey, A., Ruiz, E. and Shephard, N. (1994) Multivariate stochastic variance models. *The Review of Economic Studies*, **61**, 247–264.
- Harvey, A. C. (1990) *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.
- Henderson, H. V. and Searle, S. R. (1981) On deriving the inverse of a sum of matrices. *SIAM Review*, **23**, 53–60.
- Jacob, P. E., Lindsten, F. and Schön, T. B. (2019) Smoothing with couplings of conditional particle filters. *Journal of the American Statistical Association*, 1–20.
- Jacquier, E., Polson, N. G. and Rossi, P. E. (2002) Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, **20**, 69–87.
- Jungbacker, B. and Koopman, S. J. (2007) Monte Carlo estimation for nonlinear non-Gaussian state space models. *Biometrika*, **94**, 827–839.
- Kalman, R. E. (1960) A new approach to linear filtering and prediction problems. *American Society of Mechanical Engineers: Journal of Basic Engineering*, **82(1)**, 35–45.

- Katzfuss, M., Stroud, J. R. and Wikle, C. K. (2019) Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association*, 1–43.
- Koopman, S. J., Lucas, A. and Scharth, M. (2015) Numerically accelerated importance sampling for nonlinear non-Gaussian state-space models. *Journal of Business & Economic Statistics*, **33**, 114–127.
- (2016) Predicting time-varying parameters with parameter-driven and observation-driven models. *Review of Economics and Statistics*, **98**, 97–110.
- Kullback, S. and Leibler, R. A. (1951) On information and sufficiency. *The Annals of Mathematical Statistics*, **22**, 79–86.
- Levinson, S. E., Rabiner, L. R. and Sondhi, M. M. (1983) An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell System Technical Journal*, **62**, 1035–1074.
- Lin, M. T., Zhang, J. L., Cheng, Q. and Chen, R. (2005) Independent particle filters. *Journal of the American Statistical Association*, **100**, 1412–1421.
- Nocedal, J. and Wright, S. (2006) *Numerical Optimization*. Springer Science & Business Media.
- Polson, N. G., Stroud, J. R. and Müller, P. (2008) Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**, 413–428.
- Richard, J.-F. and Zhang, W. (2007) Efficient high-dimensional importance sampling. *Journal of Econometrics*, **141**, 1385–1411.
- Ruiz, E. (1994) Quasi-maximum likelihood estimation of stochastic volatility models. *Journal of Econometrics*, **63**, 289–306.
- Sardy, S. and Tseng, P. (2004) On the statistical analysis of smoothing by maximizing dirty Markov random field posterior distributions. *Journal of the American Statistical Association*, **99**, 191–204.
- Scott, S. L. (2002) Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, **97**, 337–351.
- Shephard, N. and Pitt, M. K. (1997) Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, **84**, 653–667.
- Sherman, J. and Morrison, W. J. (1950) Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, **21**, 124–127.
- Singh, A. and Roberts, G. (1992) State space modelling of cross-classified time series of counts. *International Statistical Review*, **60**, 321–335.
- So, M. K. (2003) Posterior mode estimation for nonlinear and non-Gaussian state space models. *Statistica Sinica*, 255–274.

- Stroud, J. R., Müller, P. and Polson, N. G. (2003) Nonlinear state-space models with state-dependent variances. *Journal of the American Statistical Association*, **98**, 377–386.
- Tauchen, G. E. and Pitts, M. (1983) The price variability-volume relationship on speculative markets. *Econometrica: Journal of the Econometric Society*, 485–505.
- Taylor, S. J. (2008) *Modelling Financial Time Series*. World Scientific.
- Weierstrass, K. (1885) Über die analytische darstellbarkeit sogenannter willkürlicher functionen einer reellen veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, **2**, 633–639.
- Yun, J., Yang, F. and Chen, Y. (2017) Augmented particle filters. *Journal of the American Statistical Association*, **112**, 300–313.