

Ancillarity-Sufficiency or not; Interweaving to Improve MCMC Estimation of the Local Level DLM

Matt Simpson

January 31, 2014

Abstract

In dynamic linear models (DLMs), MCMC sampling can often be very slow for estimating the posterior density — especially for longer time series. In particular, in some regions of the parameter space the standard data augmentation algorithm can mix very slowly. Recently ancillarity-sufficiency interweaving has been introduced as a method to take advantage of alternate parameterizations in multilevel models in order to improve the mixing and convergence properties of the chain. Focusing on the local level DLM, we explore alternate parameterizations and various interweaving algorithms through simulation in order to improve mixing. We conclude by explaining what our results may mean for MCMC in a more general DLM.

1 Model

The general dynamic linear model (DLM) is a linear, gaussian, state space model. A state space model has two components — a sequence of real valued random vectors $\{y_t\}$ denoting an observation for each period and another sequence of real valued random vectors $\{\theta_t\}$ denoting a latent state for each period. The observations range from $t = 1, \dots, T$, i.e. the length of the full time series, and the states range from $t = 0, \dots, T$. The states form a Markov chain so that $p(\theta_{t+1}|\theta_{0:T}) = p(\theta_{t+1}|\theta_t)$ where $p(x|z)$ denotes the conditional density of x given z . Furthermore, the observations are conditionally independent given the states and in particular $p(y_{1:T}|\theta_{0:T}) = p(y_1|\theta_1) \times \dots \times p(y_T|\theta_T)$. The state space model is then completed by specifying the observation and system equations: for $t = 1, 2, \dots, T$

$$y_t = f_t(\theta_t, v_t) \tag{1}$$

$$\theta_t = g_t(\theta_{t-1}, w_t) \tag{2}$$

where $v_{1:T}$ and $w_{1:T}$ are independent and are each iid draws from some distribution. Equation (1) is known as the observation equation since it describes how the observations depend on the current latent state and (2) is known the system equation since it describes how the latent states, or the underlying system, evolve over time. The random vector v_t is called the observation error and w_t is called the system error or the system disturbance. The functions f_t and g_t and the distributions of $v_{1:T}$ and $w_{1:T}$ may depend on some unknown parameter vector ϕ that we wish to estimate.

The dynamic linear model adds a couple constraints to the state space model. First, it requires that both f_t and g_t be linear functions. Second, it requires that $(v_{1:T}, w_{1:T})$ is normally distributed, usually with a mean of zero. We can then rewrite the DLM as

$$y_t|\theta_{0:T} \stackrel{ind}{\sim} N(F_t\theta_t, V_t) \tag{3}$$

$$\theta_t|\theta_{0:t-1} \sim N(G_t\theta_{t-1}, W_t) \tag{4}$$

for $t = 1, 2, \dots, T$ where F_t and G_t are matrices, and V_t and W_t are symmetric and positive definite covariance matrices. If θ_t is $p \times 1$ and y_t is $k \times 1$, then F_t is $k \times p$ and G is $p \times p$ while V_t is $k \times k$ and W_t is $p \times p$. The

observation errors (“errors”), $v_t = y_t - F_t \theta_t$ for $t = 1, 2, \dots, T$, and the system disturbances (“disturbances”), $w_t = \theta_t - G_t \theta_{t-1}$ for $t = 1, 2, \dots, T$ are independent. Let ϕ denote the unknown parameter vector. Then possibly $F_{1:T}$, $G_{1:T}$, $V_{1:T}$, and $W_{1:T}$ are all functions of ϕ . We’ll focus our attention on a simpler version of the DLM. Specifically, suppose that F_t and G_t are known matrices for $t = 1, 2, \dots, T$ and that both V_t and W_t are fully unknown for $t = 1, 2, \dots, T$. Thus $\phi = (V_{1:T}, W_{1:T})$ is our unknown parameter vector.

To complete the model specification in a Bayesian context, we need priors on θ_0 , $V_{1:T}$, and $W_{1:T}$. We’ll assume that they’re mutually independent a priori and that $\theta_0 \sim N(m_0, C_0)$, $V_t \sim IW(\Psi_t, \eta_t)$ for $t = 1, 2, \dots, T$, and $W_t \sim IW(\Omega_t, \delta_t)$ for $t = 1, 2, \dots, T$ where m_0 , C_0 and Ψ_t , η_t , Ω_t , and δ_t for $t = 1, 2, \dots, T$ are known hyperparameters and $IW(\Psi, \eta)$ denotes the inverse Wishart density with degrees of freedom η and positive definite scale matrix Ψ . This allows us to write the full joint distribution of $(V_{1:T}, W_{1:T}, \theta_{0:T}, y_{1:T})$ as

$$\begin{aligned} p(V_{1:T}, W_{1:T}, \theta_{0:T}, y_{1:T}) &\propto \exp \left[-\frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \right] \\ &\times \prod_{t=1}^T |V_t|^{-(\eta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Psi_t V_t^{-1}) \right] |V_t|^{-1/2} \exp \left[-\frac{1}{2} (y_t - F_t \theta_t)' V_t^{-1} (y_t - F_t \theta_t) \right] \\ &\times |W_t|^{-(\delta_t + p + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Omega_t W_t^{-1}) \right] |W_t|^{-1/2} \exp \left[-\frac{1}{2} (\theta_t - G_t \theta_{t-1})' W_t^{-1} (\theta_t - G_t \theta_{t-1}) \right] \end{aligned} \quad (5)$$

where $p = \dim(\theta_t)$, $k = \dim(y_t)$, and $\text{tr}(\cdot)$ is the matrix trace operator.

2 Estimating the Model via Data Augmentation: Parameterization Issues

The usual way to estimate the model is via data augmentation (DA) using forward filtering backward sampling (FFBS), as in Frühwirth-Schnatter [1994] and Carter and Kohn [1994]. The basic idea is to implement a Gibbs sampler with two blocks. The generic DA algorithm with parameter ϕ , augmented data θ , and data y obtains the $k + 1$ ’st state of the Markov chain, $\phi^{(k+1)}$, from the k ’th state, $\phi^{(k)}$ as follows:

Algorithm 1.

1. Draw θ from $p(\theta | \phi^{(k)}, y)$
2. Draw $\phi^{(k+1)}$ from $p(\phi | \theta, y)$

The first block samples the states conditional on the data and model parameters while the second block samples the parameters conditional on the states and the data. We’re calling this algorithm the “state sampler.” The FFBS step consists of running the Kalman filter to obtain a draw from $\theta_T | V_{1:T}, W_{1:T}, y_{1:T}$, then moving backward to obtain draws from $\theta_t | V_{1:T}, W_{1:T}, y_{1:T}, \theta_{t+1:T}$ for $t = T - 1, T - 2, \dots, 0$. Frühwirth-Schnatter [1994], Carter and Kohn [1994], and Petris et al. [2009] contain the details of this process. For the subset of DLMs we are considering, the algorithm cashes out like this:

Algorithm 2.

1. Draw $\theta_{0:T}$ from $p(\theta_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$ using FFBS
2. Draw $(V_{1:T}^{(k+1)}, W_{1:T}^{(k+1)})$ from $p(V_{1:T}, W_{1:T} | \theta_{0:T}, y_{1:T})$:

V_1, V_2, \dots, V_T and W_1, W_2, \dots, W_T are conditionally independent given $(\theta_{0:T}, y_{1:T})$, with distributions

$$V_t | \theta_{0:T}, y_{1:T} \sim IW(\Psi_t + v_t v_t', \eta_t + 1) \quad W_t | \theta_{0:T}, y_{1:T} \sim IW(\Omega_t + w_t w_t', \delta_t + 1)$$

where $v_t = y_t - F_t \theta_t$ and $w_t = \theta_t - G_t \theta_{t-1}$.

The main problem with this algorithm is that computation time increases quickly with the length of the time series because the Kalman filter essentially requires drawing from $\theta_t|V_{1:T}, W_{1:T}, \theta_{0:t}, y_{0:T}$ for $t = 0, 1, \dots, T$, so the FFBS step represents $2T$ multivariate draws. A second problem is that in some regions of the parameter space, the Markov chain mixes poorly for some of the parameters. For example, in the univariate local level model and similar models it's known that if the time constant variance of the latent states, W , is too small, mixing will be poor Frühwirth-Schnatter [2004].

One well known method of improving mixing and convergence in MCMC samplers is reparameterizing the model. Papaspiliopoulos et al. [2007] is a good summary. Most of the work in some way focuses on what are called centered and noncentered parameterizations. In our general notation where ϕ is the parameter, θ is the DA and y is the data, the parameterization (ϕ, θ) is a *centered parameterization* (CP) if $p(y|\theta, \phi) = p(y|\theta)$. The parameterization is a *noncentered parameterization* (NCP) if $p(\theta|\phi) = p(\theta)$. When (ϕ, θ) is a CP, θ is called a *centered augmentation* (CA) for ϕ and when (ϕ, θ) is a NCP, θ is called a *noncentered augmentation* (NCA) for ϕ . A centered augmentation is sometimes called a *sufficient augmentation* (SA) and a noncentered augmentation is sometimes called an *ancillary augmentation* (AA), e.g. in Yu and Meng [2011]. Like Yu and Meng, we prefer the latter terminology because it immediately suggests the intuition that a sufficient augmentation is like a sufficient statistic while an ancillary augmentation is like an ancillary statistic.

The key reasoning behind the emphasis on SAs and AAs is that typically when the DA algorithm based on the SA has nice mixing and convergence properties the DA algorithm based on the AA has poor mixing and convergence properties and vice versa. In other words, the two algorithms form a “beauty and the beast” pair. This property suggests that there might be some way to combine the two DA algorithms or the two underlying parameterizations in order to construct a sampler which has “good enough” properties all the time. Some work focuses on using partially noncentered parameterizations that are a sort of bridge between the CP and NCP, e.g. Papaspiliopoulos et al. for general hierarchical models and Frühwirth-Schnatter [2004] in the context of a particular DLM — a dynamic univariate regression with a stationary AR(1) coefficient. But this doesn't quite accomplish what we want because it still picks a single parameterization to use that may depend on the region of the parameter space the posterior concentrates most of its mass. The interweaving concept of Yu and Meng [2011] does precisely what we want, however. The idea is pretty simple: suppose that ϕ denotes the parameter vector, θ denotes one augmented data vector, γ denotes another augmented data vector, and y denotes the data. Then an MCMC algorithm that *interweaves* between θ and γ performs the following steps in a single iteration to obtain the $k+1$ 'st draw, $\phi^{(k+1)}$, from the k 'th draw, $\phi^{(k)}$:

Algorithm 3.

1. Draw θ from $p(\theta|\phi^{(k)}, y)$
2. Draw $\gamma^{(k+1)}$ from $p(\gamma|\theta, y)$
3. Draw $\phi^{(k+1)}$ from $p(\phi|\gamma^{(k+1)}, y)$.

Notice that an additional step is added to algorithm 1, and the final step now draws ϕ conditional on γ instead of θ . This is the intuition behind the name “interweaving”—the draw of the second augmented data vector is weaved in between the draws of θ and ϕ . This particular method of interweaving is called a *global* interweaving strategy (GIS) since interweaving occurs globally across the entire parameter vector. It's possible to define a *componentwise* interweaving strategy (CIS) that interweaves within specific steps of a Gibbs sampler as well. Step two of the GIS algorithm is typically accomplished by sampling $\phi|\theta, y$ and then $\gamma|\theta, \phi, y$. In addition, γ and θ are often, but not always, one-to-one transformations of each other conditional on (ϕ, y) , i.e. $\gamma = M(\theta; \phi, y)$. Where $M(\cdot; \phi, y)$ is a one-to-one function. In this case, the algorithm becomes:

Algorithm 4.

1. Draw θ from $p(\theta|\phi^{(k)}, y)$
2. Draw ϕ from $p(\phi|\theta, y)$
3. Draw γ from $p(\gamma|\theta, \phi, y)$

4. Draw $\phi^{(k+1)}$ from $p(\phi|\gamma, y)$

When γ is not a one-to-one transformation of θ , step 4 is an update $\gamma = M(\theta; \phi, y)$. The GIS algorithm is directly comparable to an *alternating* algorithm. Given the same two DAs, θ and γ , and parameter vector ϕ , the alternating algorithm for sampling from $p(\phi|y)$ is as follows:

Algorithm 5.

1. Draw θ from $p(\theta|\phi^{(k)}, y)$
2. Draw ϕ from $p(\phi|\theta, y)$
3. Draw γ from $p(\gamma|\phi, y)$
4. Draw $\phi^{(k+1)}$ from $p(\phi|\gamma, y)$

The key difference between this algorithm and algorithm 4 is in step 3: instead of drawing from $p(\gamma|\theta, \phi, y)$, the alternating algorithm draws from $p(\gamma|\phi, y)$. In other words it alternates between two data augmentation algorithms in a single iteration. The interweaving algorithm, on the other hand, connects or “weaves” the two separate iterations together in step 3 by drawing γ conditional on θ in addition to ϕ and y .

Yu and Meng call a GIS approach where one of the DAs is a SA and the other is an AA an *ancillary sufficient interweaving strategy*, or an ASIS. They show that the GIS algorithm has a geometric rate of convergence no worse than the worst of the two underlying algorithms and in some cases better than the the corresponding alternating algorithm. In models with a “nice” prior on ϕ in some sense, they also show that the ASIS algorithm is the same as the optimal PX-DA algorithm of Meng and Van Dyk [1999], Liu and Wu [1999], Van Dyk and Meng [2001] and Hobert and Marchev [2008]. Their results suggest that ASIS is a promising approach to improve the speed of MCMC in a variety of models no matter what region of the parameter space the posterior is concentrated. To gain some intuition about why this is so, recall that a typical problem with slow MCMC is that there is high autocorrelation in the Markov chain for ϕ , $\{\phi^{(k)}\}_{k=1}^K$, leading to imprecise estimates of $E[f(\phi)]$ for some function f . Our ultimate goal here is to reduce this dependence. In the usual DA algorithm, e.g. algorithm 1, when ϕ and θ are highly dependent in the joint posterior the draws from $p(\theta|\phi, y)$ and then from $p(\phi|\theta, y)$ won’t move the chain much, resulting in high autocorrelation in the chain. Interweaving helps break this autocorrelation in two ways. First, by inserting the extra step, e.g. steps 2 and 3 together in 4, the chain gets an additional chance to move in a single iteration thereby weakening the autocorrelation. Second, when one of θ and γ is a “beauty” and the other is a “beast”, as is often the case when they form a SA-AA pair, one of steps 2 and 4 in algorithm 4 will significantly move the chain even if the other step will not. This intuition suggests that the key isn’t so much that θ and γ form a SA-AA pair as that they form a beauty and the beast pair. It just so happens that SA-AA pairs are often great at accomplishing this.

2.1 The Scaled Disturbances

The next step is to apply the ideas of interweaving to sampling from the posterior of the dynamic linear model. Papaspiliopoulos et al. note that typically the usual parameterization results in a SA for the parameter ϕ . All that’s necessary for an ASIS algorithm, then, is to construct an AA for ϕ . We immediately run into a problem because the standard DA for a DLM is the latent states $\theta_{0:T}$. From equations (3) and (4) we see that $V_{1:T}$ is in the observation equation so that $\theta_{0:T}$ isn’t a SA for $(V_{1:T}, W_{1:T})$ while $W_{1:T}$ is in the system equation so that $\theta_{0:T}$ isn’t an AA for $(V_{1:T}, W_{1:T})$ either. In order to find a SA we need to somehow move $V_{1:T}$ from the observation equation (3) to the system equation (4) while leaving $W_{1:T}$ in the system equation. Alternatively, we could find an AA by somehow moving $W_{1:T}$ from the system equation to the observation equation while leaving $V_{1:T}$ in the observation equation. A naive thing to try is to condition on the disturbances instead of the states and see if the disturbances for a SA or an AA for $(V_{1:T}, W_{1:T})$. The disturbances $w_{0:T}$ are defined by $w_t = \theta_t - G_t\theta_{t-1}$ for $t = 0, 1, \dots, T$ and we define $\theta_{-1} = 0$ so that $w_0 = \theta_0$. However the DA algorithm based on the w_t ’s is identical to the algorithm based on the θ_t ’s. This is because

$w_{0:T}$ is a one-to-one function of $\theta_{0:T}$ that doesn't depend on $V_{1:T}$ or $W_{1:T}$, the conditional distributions $p(V_{1:T}, W_{1:T} | \theta_{0:T}, y_{1:T})$ and $p(V_{1:T}, W_{1:T} | w_{0:T}, y_{1:T})$ are identical.

Papaspiliopoulos et al. suggest that in order to obtain an ancillary augmentation for a variance parameter, we must scale the sufficient agumentation by the square root of that parameter. Based on this intuition, note that if we hold $V_{1:T}$ constant then $\theta_{0:T}$ is a SA for $W_{1:T}$ from the observation and system equations, (3) and (4), i.e. we say $\theta_{0:T}$ is a SA for $W_{1:T}$ given $V_{1:T}$, or for $W_{1:T} | V_{1:T}$. Similarly $\theta_{0:T}$ is an AA for $V_{1:T} | W_{1:T}$. This suggests that if we scale θ_t by W_t for all t appropriately we'll have an ancillary augmentation for $V_{1:T}$ and $W_{1:T}$ jointly. The same intuition suggests scaling $w_t = \theta_t - G_t \theta_{t-1}$ by W_t for all t appropriately in order to find an ancillary augmentation for $(V_{1:T}, W_{1:T})$. We'll work with the latter case though, again these two ideas are ultimately equivalent since the resulting DAs are one-to-one transformations of each other.

To start with, we'll rewrite (5) in a less messy form. Define the following matrices:

$$V = \begin{bmatrix} I_k & 0 & \cdots & 0 \\ 0 & V_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_T \end{bmatrix}, \quad W = \begin{bmatrix} C_0 & 0 & \cdots & 0 \\ 0 & W_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_T \end{bmatrix},$$

$$G = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ G_1 & 0 & \cdots & 0 & 0 \\ 0 & G_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & G_T & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & F_1 & 0 & \cdots & 0 \\ 0 & 0 & F_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & F_T \end{bmatrix},$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_T \end{bmatrix}, \quad y = \begin{bmatrix} 0 \\ y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix}, \quad \text{and } m = \begin{bmatrix} m_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where I_k is the $k \times k$ identity matrix. With the notation so defined, we can write the full joint distribution of data and parameters as

$$p(V_{1:T}, W_{1:T}, \theta_{0:T}, y_{1:T}) \propto p(V_{1:T})p(W_{1:T}) \times |V|^{-1/2} \exp \left[-\frac{1}{2} (y - F\theta)' V^{-1} (y - F\theta) \right] |W|^{-1/2} \exp \left[-\frac{1}{2} (\theta - G\theta)' W^{-1} (\theta - G\theta) \right] \quad (6)$$

Now we can define the scaled disturbances in the general DLM. Let L_t denote the Cholesky decomposition of W_t^{-1} , i.e. $L_t' L_t = W_t^{-1}$, and define

$$L = \begin{bmatrix} L_0 & 0 & \cdots & 0 \\ 0 & L_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_T \end{bmatrix}.$$

So we have $L' L = W^{-1}$. Then we'll define the scaled disturbances as $\gamma = L[(I_{(T+1)p} - G)\theta - m]$ where again I_n is the $n \times n$ identity matrix. In other words $\gamma_0 = \theta_0 - m_0$ and for $t = 1, 2, \dots, T$, $\gamma_t = L_t(\theta_t - G_t \theta_{t-1})$. We can confirm our intuition that the scaled disturbances are an AA for $V_{1:T}$ and $W_{1:T}$ jointly. The reverse transformation is $\theta = (I_{(T+1)p} - G)^{-1} (L^{-1} \gamma + m)$. Then the Jacobian is $J = (I_{(T+1)p} - G)^{-1} L^{-1}$ so $|J| = |L|^{-1} |I_{(T+1)p} - G|^{-1} = |W|^{-1/2}$ since $I_{(T+1)p} - G$ is block lower triangular with identity matrices in

the diagonal blocks. Then from (6) we can write the full joint distribution of $(V_{1:T}, W_{1:T}, \gamma_{0:T}, y_{1:T})$ as

$$p(V_{1:T}, W_{1:T}, \gamma_{0:T}, y_{1:T}) \propto p(V_{1:T})p(W_{1:T}) \exp \left[-\frac{1}{2} \gamma' \gamma \right] \\ \times \exp \left[-\frac{1}{2} [y - F(I_{(T+1)p} - G)^{-1}(L^{-1}\gamma + m)]' V^{-1} [y - F(I_{(T+1)p} - G)^{-1}(L^{-1}\gamma + m)] \right]. \quad (7)$$

So ultimately under the scaled disturbance parameterization we can write the model as

$$y_t | \gamma_{0:T}, V_{1:T}, W_{1:T} \stackrel{ind}{\sim} N \left(F_t \sum_{s=0}^t H_{s+1}^t L_s^{-1} \gamma_s, V_t \right) \\ \gamma_t \stackrel{iid}{\sim} N(0, I_p)$$

for $t = 1, 2, \dots, T$ where I_p is the $p \times p$ identity matrix. Neither $V_{1:T}$ nor $W_{1:T}$ are in the system equation, so the scaled disturbances are an AA for $(V_{1:T}, W_{1:T})$. In fact, Frühwirth-Schnatter [2004] use this parameterization in a dynamic regression model with stationary regression coefficient and also note that it is a noncentered parameterization (NCP), i.e. an AA for $(V_{1:T}, W_{1:T})$ in our terminology.

The DA algorithm base on $\gamma_{0:T}$ is as follows:

Algorithm 6.

1. Draw $\gamma_{0:T}$ from $p(\gamma_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$
2. Draw $(V_{1:T}^{(k+1)}, W_{1:T}^{(k+1)})$ from $p(V, W | \gamma_{0:T}, y_{1:T})$.

Step 1 can be accomplished directly with the disturbance smoother of Koopman [1993] or indirectly by using FFBS to draw the states and then transform them to the scaled disturbances. Step 2 ends up being complicated because the joint conditional posterior of V and W isn't a known density. We'll go through an example of this when both y_t and θ_t are scalars later.

We previously mentioned that the intuition behind the scaled disturbances also suggests trying the scaled states, i.e. $\theta_{0:T}^s$ where $\theta_0^s = \theta_0$ and for $t = 1, 2, \dots, T$, $\theta_t^s = L_t \theta_t$. Note that $\theta_{0:T}^s$ and $\gamma_{0:T}$ are completely determined by each other. This suggests (but doesn't imply due to the Borel–Kolmogorov paradox) that the conditional posterior of $(V_{1:T}, W_{1:T})$ is the same whether we condition on $\gamma_{0:T}$ or $\theta_{0:T}^s$. In fact, this is true and easy to verify as well is the fact that $\theta_{0:T}^s$ is also an AA for $(V_{1:T}, W_{1:T})$. As such, we have a choice to make between two equivalent representations of what amounts to the same parameterization. We prefer the scaled disturbances ($\gamma_{0:T}$) to the scaled states ($\theta_{0:T}^s$) for two reasons. First, the scaled disturbances are already in the literature, e.g. in Frühwirth-Schnatter [2004] and are a natural extension of the concept of noncentering as it is typically used in hierarchical models to time series models (e.g. in Van Dyk and Meng [2001] where the suggestion is to demean before scaling). Second, the scaled disturbances immediately suggest another potential AA that seems like it should be analogous — the scaled observation errors, or more succinctly the scaled errors. That is, $v_t = y_t - F_t \theta_t$ appropriately scaled by V_t in the general DLM.

2.2 The Scaled Errors

Define the scaled errors as $\psi_t = K_t(y_t - F_t \theta_t)$ for $t = 1, 2, \dots, T$ and $\psi_0 = \theta_0$ where K_t is the cholesky decomposition of V_t , that is $K_t' K_t = V_t$. This is a bit strange since in general $\dim(\psi_0) \neq \dim(\psi_t)$ for $t = 1, 2, \dots, T$. Ideally we might like an “ F_0 ” so that we can set $\psi_0 = F_0 \theta_0$ in order for ψ_0 to have the same dimension as ψ_1 . In general, though, there is no F_0 . In some DLMS F_t is constant with respect to t so that we could set $F_0 = F$, but in dynamic regression for example, F_0 will be a different value of all covariates and may be unknown or be an essentially arbitrary choice. The arbitrariness is fine since it is just a definition, but $\psi_0 = \theta_0$ seems most natural in the general context.

With this definition of $\psi_{0:T}$, it isn't straightforward to determine $p(\psi_{1:T} | V_{1:T}, W_{1:T})$, i.e. to write down the model in terms of $\psi_{0:T}$ instead of $\theta_{0:T}$. When F_t is $k \times k$ (so that $\dim(y_t) = k = p = \dim(\theta_t)$) and is invertible for $t = 1, 2, \dots, T$, the $\psi_{0:T}$ is a one-to-one transformation of $\theta_{0:T}$ and the problem is easier. Then

$\theta_t = F_t^{-1}(y_t - K_t^{-1}\psi_t)$ for $t = 1, 2, \dots, T$ while $\theta_0 = \psi_0$. The Jacobian of this transformation is block diagonal with a single copy of the identity matrix and the $F_t^{-1}K_t^{-1}$'s along the diagonal, so $|J| = \prod_{t=1}^T |F_t|^{-1} |V_t|^{-1/2}$. So from (5) we can write the joint distribution of $(V_{1:T}, W_{1:T}, \psi_{0:T}, y_{1:T})$ as

$$\begin{aligned} p(V_{1:T}, W_{1:T}, \psi_{0:T}, y_{1:T}) &\propto \exp \left[-\frac{1}{2}(\psi_0 - m_0)' C_0^{-1}(\psi_0 - m_0) \right] \\ &\times \prod_{t=1}^T |V_t|^{-(\eta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Psi_t V_t^{-1}) \right] \exp \left[-\frac{1}{2} \psi_t' \psi_t \right] \\ &\times |W_t|^{-(\delta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Omega_t W_t^{-1}) \right] |W_t|^{-1/2} \exp \left[-\frac{1}{2} (y_t - \mu_t)' (F_t W_t F_t')^{-1} (y_t - \mu_t) \right] \end{aligned} \quad (8)$$

where we define $\mu_t = K_t^{-1}\psi_t + F_t G_t F_{t-1}^{-1}(y_{t-1} - K_{t-1}^{-1}\psi_{t-1})$, $y_0 = 0$, $K_0 = I_k$, and $F_0 = I_k$ where I_k is the $k \times k$ identity matrix. Now we can write the model in terms of the scaled error parameterization:

$$\begin{aligned} y_t | V_{1:T}, W_{1:T}, \psi_{0:T}, y_{1:t-1} &\sim N(\mu_t, F_t' W_t F_t) \\ \psi_t &\stackrel{iid}{\sim} N(0, I_k) \end{aligned}$$

for $t = 1, 2, \dots, T$. Now we see immediately that the scaled errors, $\psi_{0:T}$, are also an AA for $(V_{1:T}, W_{1:T})$ since neither V nor W are in the system equation of this model, though note that both $V_{1:T}$ and $W_{1:T}$ are in the observation equation, so $\psi_{0:T}$ is not a SA for $(V_{1:T}, W_{1:T})$ or for either one given the other.

The DA algorithm based on $\psi_{0:T}$ is:

Algorithm 7.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$
2. Draw $(V_{1:T}^{(k+1)}, W_{1:T}^{(k+1)})$ from $p(V_{1:T}, W_{1:T} | \psi_{0:T}, y_{1:T})$.

Once again step one can be accomplished directly with Koopman's disturbance smoother or indirectly using FFBS. Step 2 is also once again complicated since the joint conditional posterior of $V_{1:T}$ and $W_{1:T}$ isn't a known density.

2.3 Interweaving in the DLM: Global and Componentwise

We now have three DAs for the generic DLM with known F_t 's and G_t 's. For simplicity we'll assume that $\dim(y_t) = \dim(\theta_t)$ and F_t invertible for $t = 1, 2, \dots, T$ so that the scaled errors are easy to work with. The three DAs are the states, $\theta_{0:T}$, the scaled disturbances $\gamma_{0:T}$, and the scaled errors $\psi_{0:T}$. This allows to construct four separate GIS algorithms based on algorithm 4: three algorithms that interweave between any two of $\theta_{0:T}$, $\gamma_{0:T}$, and $\psi_{0:T}$, the state-dist, state-error, and dist-error interweaving algorithms, and one algorithm that interweaves between all three, the triple interweaving algorithm. Note that strictly speaking, the order in which we sample the DAs in the algorithm does matter, but Yu and Meng note that this tends to make very little difference so while we actually have twelve separate GIS samplers (two of each GIS sampler depending on two DAs, and six GIS samplers depending on all three), effectively we only have four. For example, Algorithm 8 is the state-dist GIS algorithm, for example:

Algorithm 8.

1. Draw $\theta_{0:T}$ from $p(\theta_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$
2. Draw $(V_{1:T}^{(k+0.5)}, W_{1:T}^{(k+0.5)})$ from $p(V_{1:T}, W_{1:T} | \theta_{0:T}, y_{1:T})$
3. Update $\gamma_{0:T}^{(k+1)}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - G_t \theta_{t-1})$ for $t = 1, 2, \dots, T$

4. Draw $(V_{1:T}^{(k+1)}, W_{1:T}^{(k+1)})$ from $p(V_{1:T}, W_{1:T} | \gamma_{0:T}, y_{1:T})$

where again $L_t^{(k+0.5)}$ is the Cholesky decomposition of $(W_t^{(k+0.5)})^{-1}$, i.e. $(L_t^{(k+0.5)})' L_t^{(k+0.5)} = (W_t^{(k+0.5)})^{-1}$. Steps 1 and 2 are the same as steps 1 and 2 in algorithm 2, and step 4 is the same as step 2 of algorithm 6. The triple interweaving algorithm is the same as algorithm 8 except it adds two more steps at the end: an update of $\psi_{0:T}$ from $\gamma_{0:T}$ and the draw of $(V_{1:T}, W_{1:T})$ in step 4, and then a draw from $(V_{1:T}, W_{1:T} | \psi_{0:T}, y_{1:T})$. In practice we may want to break up step 4 into two steps if it's easier to draw from the full conditionals of $V_{1:T}$ and $W_{1:T}$ rather than drawing them jointly. Algorithm 9 below is exactly this.

Algorithm 9.

1. Draw $\theta_{0:T}$ from $p(\theta_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$
2. Draw $(V_{1:T}, W_{1:T})$ from $p(V_{1:T}, W_{1:T} | \theta_{0:T}, y_{1:T})$
3. Update $\gamma_{0:T}^{(k+1)}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t(\theta_t - G_t \theta_{t-1})$ for $t = 1, 2, \dots, T$
4. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T} | W_{1:T}, \gamma_{0:T}, y_{1:T})$
5. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T} | V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$

Step 4 actually draws $V_{1:T}$ from the same density as in step 2, but only the last of the two draws is used for anything in the algorithm. To see this, note that $V_{1:T}$ and $W_{1:T}$ are independent in their conditional posterior given $\theta_{0:T}$ with each V_t and W_t having inverse Wishart distributions given in Algorithm 2. In particular, $p(V_{1:T} | W_{1:T}^{(k)}, \theta_{0:T}, y_{1:T})$ is the product of independent inverse Wishart distributions with parameters $\Psi_t + v_t v_t'$ and $\eta_t + 1$ with $v_t = y_t - F_t \theta_t$. But $p(V_{1:T} | W_{1:T}^{(k)}, \gamma_{0:T}, y_{1:T})$ is identical to this since by (7):

$$p(V_{1:T} | W_{1:T}, \gamma_{0:T}, y_{1:T}) \propto p(V_{1:T}, W_{1:T}, \gamma_{0:T}, y_{1:T}) \propto \prod_{t=1}^T |V_t|^{-(\eta_t + k + 3)/2} \exp \left[-\frac{1}{2} \text{tr} ((\Psi_t + \nu_t \nu_t') V_t^{-1}) \right]$$

where $\nu_t = y_t - F_t \sum_{s=0}^t H_{s+1}^t (L_s^{(k+0.5)})^{-1} \gamma_s$. In other words, independent inverse Wishart distributions with parameters $\Psi_t + \nu_t \nu_t'$ and $\eta_t + 1$. Now $\theta_t = \sum_{s=0}^t H_{s+1}^t (L_s^{(k+0.5)})^{-1} \gamma_s$ because this is just the back transform of step 4 of Algorithm 10 or step 3 of Algorithm 8, so $\nu_t = v_t$ and thus the distributions are the same. As a result, we can either only draw $W_{1:T}$ in step 2 or step 4 can be omitted.

Note that none of these GIS algorithms are ASIS algorithms — none of the DAs are a SA for $(V_{1:T}, W_{1:T})$. The states, $\theta_{0:T}$, are a SA for $W_{1:T} | V_{1:T}$ though, so this motivates a CIS algorithm. A partial CIS algorithm is immediate:

Algorithm 10.

1. Draw $\theta_{0:T}$ from $p(\theta_{0:T} | V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$
2. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T} | W_{1:T}^{(k)}, \theta_{0:T}, y_{1:T})$
3. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T} | V_{1:T}^{(k+1)}, \theta_{0:T}, y_{1:T})$
4. Update $\gamma_{1:T}^{(k+1)}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - G_t \theta_{t-1})$ for $t = 1, 2, \dots, T$
5. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T} | V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$

This algorithm is actually the same as a version of the state-dist interweaving algorithm, specifically Algorithm 9. So we can construct a partial CIS algorithm, but it's actually the exact same algorithm as a GIS algorithm.

With a little more work, we can also construct a full CIS algorithm that also turns out to be the same as another GIS algorithm. Recall that $\gamma_t = L_t(\theta_t - \theta_{t-1})$ and $\psi_t = K_t(y_t - \theta_t)$ for $t = 1, 2, \dots, T$ where

$L_t' L_t = W_t^{-1}$ and $K_t' K_t = V_t^{-1}$. Now define $\tilde{\gamma}_t = K_t(\theta_t - \theta_{t-1})$ and $\tilde{\psi}_t = L_t(y_t - \theta_t)$ for $t = 1, 2, \dots, T$ and $\tilde{\psi}_0 = \tilde{\gamma}_0 = \theta_0$. In other words, the “tilde” versions of the scaled disturbances and the scaled errors are scaled by the “wrong” standard deviation. Now we’ll show that $\gamma_{0:T}$ and $\tilde{\gamma}_{0:T}$ are an AA-SA pair for $W_{1:T}|V_{1:T}$ while $\psi_{0:T}$ and $\tilde{\psi}_{0:T}$ are an AA-SA pair for $V_{1:T}|W_{1:T}$. We’ve already shown that both $\psi_{0:T}$ and $\gamma_{0:T}$ are AAs for $(V_{1:T}, W_{1:T})$, so we just need to show that $gan\tilde{ma}_{0:T}$ is a SA for $W_{1:T}|V_{1:T}$ and that $\tilde{\psi}_{0:T}$ is a SA for $V_{1:T}|W_{1:T}$.

First consider $\tilde{\gamma}_{0:T}$. If we define $L_0 = K_0 = I_k$ where I_k is the $k \times k$ identity matrix, then $\tilde{\gamma}_t = K_t L_t^{-1} \gamma_t$ for $t = 0, 1, 2, \dots, T$. The reverse transformation is then $\gamma_t = L_t K_t^{-1} \tilde{\gamma}_t$. The Jacobian is then block diagonal with $L_t K_t^{-1}$ ’s along the diagonal. Thus $|J| = \prod_{t=0}^T |L_t| |K_t|^{-1} = \prod_{t=1}^T |W_t|^{-1/2} |V_t|^{1/2}$. Then from (7) we can write the joint distribution of $(V_{1:T}, W_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T})$ as

$$\begin{aligned} p(V_{1:T}, W_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T}) &\propto \exp \left[-\frac{1}{2} (\tilde{\gamma}_0 - m_0)' C_0^{-1} (\tilde{\gamma}_0 - m_0) \right] \prod_{t=1}^T |V_t|^{-(\eta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Psi_t V_t^{-1}) \right] \\ &\times |W_t|^{-1/2} \exp \left[-\frac{1}{2} \left(y_t - F_t \sum_{s=0}^t H_{s+1}^t K_s^{-1} \tilde{\gamma}_s \right)' V_t^{-1} \left(y_t - F_t \sum_{s=0}^t H_{s+1}^t K_s^{-1} \tilde{\gamma}_s \right) \right] \\ &\times |W_t|^{-(\delta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Omega_t W_t^{-1}) \right] \exp \left[-\frac{1}{2} \tilde{\gamma}_t' (K_t W_t K_t')^{-1} \tilde{\gamma}_t \right] \end{aligned} \quad (9)$$

where again $H_s^t = G_t G_{t-1} \dots G_s$ for $s = 1, 2, \dots, t$ and $H_{t+1}^t = I_k$. Then under $\tilde{\gamma}_{0:T}$ we can write the model as

$$\begin{aligned} y_t | \tilde{\gamma}_{0:T}, V_{1:T}, W_{1:T} &\stackrel{\text{ind}}{\sim} N \left(F_t \sum_{s=0}^t H_{s+1}^t K_s^{-1} \tilde{\gamma}_s, V_t \right) \\ \tilde{\gamma}_t &\stackrel{\text{ind}}{\sim} N(0, K_t W_t K_t') \end{aligned}$$

for $t = 1, 2, \dots, T$. Since K_t is the Cholesky decomposition of V_t , the observation equation doesn’t contain W_t . So $\tilde{\gamma}_{0:T}$ is a SA for $W_{1:T}|V_{1:T}$ and thus $\gamma_{0:T}$ and $\tilde{\gamma}_{0:T}$ form an AA-SA pair for $W_{1:T}|V_{1:T}$. Note also that since W_t and K_t are both in the system equation, $\tilde{\gamma}_{0:T}$ is not an AA for $V_{1:T}$ nor for $W_{1:T}$.

Now consider $\tilde{\psi}_t = L_t K_t^{-1} \psi_t$ for $t = 0, 1, 2, \dots, T$ where, again, $L_0 = K_0 = I_k$, the $k \times k$ identity matrix. Then $\psi_t = K_t L_t^{-1} \tilde{\psi}_t$ and the Jacobian is block diagonal with $K_t L_t^{-1}$ ’s along the diagonal. So $|J| = \prod_{t=1}^T |V_t|^{-1/2} |W_t|^{1/2}$ and from (8) we can write the joint distribution of $(V_{1:T}, W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T})$ as

$$\begin{aligned} p(V_{1:T}, W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T}) &\propto \exp \left[-\frac{1}{2} (\tilde{\psi}_0 - m_0)' C_0^{-1} (\tilde{\psi}_0 - m_0) \right] \\ &\times \prod_{t=1}^T |V_t|^{-(\eta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Psi_t V_t^{-1}) \right] \exp \left[-\frac{1}{2} \tilde{\psi}_t' (L_t V_t L_t')^{-1} \tilde{\psi}_t \right] \\ &\times |W_t|^{-(\delta_t + k + 2)/2} \exp \left[-\frac{1}{2} \text{tr} (\Omega_t W_t^{-1}) \right] |V_t|^{-1/2} \exp \left[-\frac{1}{2} (y_t - \tilde{\mu}_t)' (F_t W_t F_t')^{-1} (y_t - \tilde{\mu}_t) \right] \end{aligned} \quad (10)$$

where we define $\tilde{\mu}_t = L_t^{-1} \psi_t + F_t G_t F_{t-1} (y_{t-1} - L_{t-1}^{-1} \tilde{\psi}_{t-1})$ with, again, $y_0 = 0$ and $L_0 = K_0 = I_k$. In terms of $\tilde{\psi}_{0:T}$, the model is then:

$$\begin{aligned} y_t | V_{1:T}, W_{1:T}, \tilde{\psi}_{0:T}, y_{1:t-1} &\sim N(\tilde{\mu}_t, F_t' W_t F_t) \\ \tilde{\psi}_t &\stackrel{\text{ind}}{\sim} N(0, L_t V_t L_t') \end{aligned}$$

for $t = 1, 2, \dots, T$. Since $\tilde{\mu}_t$ only depends on W_t (through L_t) and not on V_t , $V_{1:T}$ is absent from the observation equation. Thus $\tilde{\psi}_{0:T}$ is a SA for $V_{1:T}|W_{1:T}$ and as a result $\psi_{0:T}$ and $\tilde{\psi}_{0:T}$ form an AA-SA pair for $V_{1:T}|W_{1:T}$. Note again that both W_t and V_t are in the system equation so $\tilde{\psi}_{0:T}$ is not an AA for either $V_{1:T}$ or $W_{1:T}$.

Now we can construct a full CIS algorithm:

Algorithm 11.

1. Draw $\tilde{\psi}_{0:T}$ from $p(\tilde{\psi}_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $V_{1:T}^{(k+0.5)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \tilde{\psi}_{0:T}, y_{1:T})$
3. Update $\psi_{0:T}$ from $\psi_0 = \tilde{\psi}_0$ and $\psi_t = K_t^{(k+0.5)}(L_t^{(k)})^{-1}\tilde{\psi}_t$ for $t = 1, 2, \dots, T$.
4. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \psi_{0:T}, y_{1:T})$.
5. Update $\tilde{\gamma}_{0:T}$ from $\psi_{0:T}$, $W_{1:T}^{(k)}$, and $V_{1:T}^{(k+1)}$.
6. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \tilde{\gamma}_{0:T}, y_{1:T})$
7. Update $\gamma_{0:T}$ from $\gamma_0 = \tilde{\gamma}_0$ and $\gamma_t = L_t^{(k+0.5)}(K_t^{(k+1)})^{-1}\tilde{\gamma}_t$ for $t = 1, 2, \dots, T$.
8. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$

Steps 1-4 constitute a Gibbs step for $V_{1:T}$ and steps 5-8 constitute a Gibbs step for $W_{1:T}$. Steps 1 can be accomplished by using FFBS to draw the states and transforming appropriately, or by using the disturbance smoother of Koopman [1993] and again transforming appropriately. Note that $L_t^{(k)}$ is the Cholesky decomposition of $(W_t^{(k)})^{-1}$ and $K_t^{(k)}$ is the Cholesky decomposition of $(V_t^{(k)})^{-1}$.

It turns out that $p(W_{1:T}|V_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T})$ and $p(W_{1:T}|V_{1:T}, \theta_{0:T}, y_{1:T})$ are the same density, and $p(V_{1:T}|W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T})$ and $p(V_{1:T}|W_{1:T}^{(k+1)}, \theta_{0:T}, y_{1:T})$ are also the same density. The upshot is that step 1 can be replaced with a draw from $p(\theta_{0:T}|V_{1:T}, W_{1:T}, y_{1:T})$, and any time we condition on one of the “tilde” variables, we can condition on $\theta_{0:T}$ instead. To see why this is true, first consider $p(W_{1:T}|V_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T})$. From (9) we have

$$\begin{aligned}
p(W_{1:T}|V_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T}) &\propto p(V_{1:T}, W_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T}) \\
&\propto \prod_{t=1}^T |W_t|^{-(\delta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} (\Omega_t W_t^{-1}) \right] \exp \left[-\frac{1}{2} \tilde{\gamma}_t' (K_t W_t K_t')^{-1} \tilde{\gamma}_t \right] \\
&\propto \prod_{t=1}^T |W_t|^{-(\delta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} ((\Omega_t + K_t \tilde{\gamma}_t \tilde{\gamma}_t' K_t') W_t^{-1}) \right] \\
&\propto \prod_{t=1}^T |W_t|^{-(\delta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} ((\Omega_t + w_t w_t') W_t^{-1}) \right]
\end{aligned}$$

since $\tilde{\gamma}_t = K_t L_t^{-1} \gamma_t = K_t (\theta_t - \theta_{t-1}) = K_t w_t$. In other words $W_t|V_{1:T}, \tilde{\gamma}_{0:T}, y_{1:T} \stackrel{\text{ind}}{\sim} IW(\Omega_t + w_t w_t', \delta_t + 1)$, which is the same as the distribution of $W_{1:T}|V_{1:T}, \theta_{0:T}, y_{1:T}$ in Algorithm 2.

Similarly, by (10) we have

$$\begin{aligned}
p(V_{1:T}|W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T}) &\propto p(V_{1:T}, W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T}) \\
&\propto \prod_{t=1}^T |V_t|^{-(\eta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} (\Psi_t V_t^{-1}) \right] \exp \left[-\frac{1}{2} \tilde{\psi}_t' (L_t V_t L_t')^{-1} \tilde{\psi}_t \right] \\
&\propto \prod_{t=1}^T |V_t|^{-(\eta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} ((\Psi_t + L_t \tilde{\psi}_t \tilde{\psi}_t' L_t') V_t^{-1}) \right] \\
&\propto \prod_{t=1}^T |V_t|^{-(\eta_t+k+3)/2} \exp \left[-\frac{1}{2} \text{tr} ((\Psi_t + v_t v_t')) \right]
\end{aligned}$$

since $\tilde{\psi}_t = L_t K_t^{-1} \psi_t = L_t (y_t - \theta_t) = L_t v_t$. In other words, $V_t|W_{1:T}, \tilde{\psi}_{0:T}, y_{1:T} \stackrel{\text{ind}}{\sim} IW(\Psi_t + v_t v_t', \eta_t + 1)$, which is again the same as the distribution of $V_{1:T}|W_{1:T}, \theta_{0:T}, y_{1:T}$ in Algorithm 2.

Now we can rewrite the full CIS algorithm in terms of $\theta_{0:T}$ instead of the tilde variables. We'll also rearrange the order in which $\theta_{0:T}$ and $\psi_{0:T}$ are used in the Gibbs step for $V_{1:T}$. This rearranging does change the algorithm, but it's still a full CIS algorithm.

Algorithm 12.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $V_{1:T}^{(k+0.5)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \psi_{0:T}, y_{1:T})$.
3. Update $\theta_{0:T}$ from $\theta_0 = \psi_0$ and $\theta_t = y_t - (K_t^{(k+0.5)})^{-1}\psi_t$ for $t = 1, 2, \dots, T$.
4. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \theta_{0:T}, y_{1:T})$.
5. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \theta_{0:T}, y_{1:T})$.
6. Update $\gamma_{0:T}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - \theta_{t-1})$ for $t = 1, 2, \dots, T$.
7. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$.

Note that there is no update step between the two Gibbs steps for $V_{1:T}$ and $W_{1:T}$, i.e. between steps 4 and 5, because the DA is already in the proper form to draw $W_{1:T}^{(k+0.5)}$ in step 5. The only thing left to show now is that this is the same as a GIS algorithm. Consider the error-dist GIS algorithm that interweaves between the scaled errors and the scaled disturbances:

Algorithm 13.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $(V_{1:T}^{(k+0.5)}, W_{1:T}^{(k+0.5)})$ from $p(V_{1:T}, W_{1:T}|\psi_{0:T}, y_{1:T})$.
3. Update $\gamma_{0:T}$ from $\gamma_0 = \psi_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - \theta_{t-1})$ for $t = 1, 2, \dots, T$ for $\theta_0 = \psi_0$ and $\theta_t = y_t - (K_t^{(k+0.5)})^{-1}\psi_t$ for $t = 1, 2, \dots, T$.
4. Draw $(V_{1:T}^{(k+1)}, W_{1:T}^{(k+1)})$ from $p(V_{1:T}, W_{1:T}|\gamma_{0:T}, y_{1:T})$.

This algorithm samples $V_{1:T}$ and $W_{1:T}$ jointly in steps 2 and 4. If we instead sample them from each of their full conditionals, we get another variant of this algorithm:

Algorithm 14.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $V_{1:T}^{(k+0.5)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \psi_{0:T}, y_{1:T})$.
3. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T}|V_{1:T}^{(k+0.5)}, \psi_{0:T}, y_{1:T})$.
4. Update $\gamma_{0:T}$ from $\gamma_0 = \psi_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - \theta_{t-1})$ for $t = 1, 2, \dots, T$ with $\theta_0 = \psi_0$ and $\theta_t = y_t - (K_t^{(k+0.5)})^{-1}\psi_t$ for $t = 1, 2, \dots, T$.
5. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T}|W_{1:T}^{(k+0.5)}, \gamma_{0:T}, y_{1:T})$.
6. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$.

Step 4 can be broken up into two steps: a transformation from $\psi_{0:T}$ to $\theta_{0:T}$, and another transformation from $\theta_{0:T}$ to $\gamma_{0:T}$. This allows us to rewrite Algorithm 14 as:

Algorithm 15.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $V_{1:T}^{(k+0.5)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \psi_{0:T}, y_{1:T})$.
3. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T}|V_{1:T}^{(k+0.5)}, \psi_{0:T}, y_{1:T})$.
4. Update $\theta_{0:T}$ from $\theta_0 = \psi_0$ and $\theta_t = y_t - (K_t^{(k+0.5)})^{-1}\psi_t$ for $t = 1, 2, \dots, T$.
5. Update $\gamma_{0:T}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - \theta_{t-1})$ for $t = 1, 2, \dots, T$.
6. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T}|W_{1:T}^{(k+0.5)}, \gamma_{0:T}, y_{1:T})$.
7. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$.

Now note that the draw of $W_{1:T}^{(k+0.5)}$ in step 3 actually only depends on $\theta_{0:T}$ from step 4. Similarly the draw of $V_{1:T}^{(k+1)}$ in step 6 only depends on $\theta_{0:T}$ from step 4 (and $y_{1:T}$). The latter we've already established while discussing Algorithm 9 when we showed that $p(V_{1:T}|W_{1:T}, \gamma_{0:T}, y_{1:T})$ and $p(V_{1:T}|W_{1:T}, \theta_{0:T}, y_{1:T})$ are identical. To show the former, we need to show that $p(W_{1:T}|V_{1:T}, \psi_{0:T}, y_{1:T})$ and $p(W_{1:T}|V_{1:T}, \theta_{0:T}, y_{1:T})$ are identical, which is analogous (note that when we condition on $\theta_{0:T}$, $V_{1:T}$ and $W_{1:T}$ are independent in the posterior). The upshot is that we can rearrange some of the steps without changing the algorithm. In particular, the order in which we draw $W_{1:T}^{(k+0.5)}$ and $V_{1:T}^{(k+1)}$ does not matter since they both depend only on $\theta_{0:T}$ and $y_{1:T}$, but not on each other. This allows us to rewrite Algorithm 15 as:

Algorithm 16.

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T}|V_{1:T}^{(k)}, W_{1:T}^{(k)}, y_{1:T})$.
2. Draw $V_{1:T}^{(k+0.5)}$ from $p(V_{1:T}|W_{1:T}^{(k)}, \psi_{0:T}, y_{1:T})$.
3. Update $\theta_{0:T}$ from $\theta_0 = \psi_0$ and $\theta_t = y_t - (K_t^{(k+0.5)})^{-1}\psi_t$ for $t = 1, 2, \dots, T$.
4. Draw $V_{1:T}^{(k+1)}$ from $p(V_{1:T}|\theta_{0:T}, y_{1:T})$.
5. Draw $W_{1:T}^{(k+0.5)}$ from $p(W_{1:T}|\theta_{0:T}, y_{1:T})$.
6. Update $\gamma_{0:T}$ from $\gamma_0 = \theta_0$ and $\gamma_t = L_t^{(k+0.5)}(\theta_t - \theta_{t-1})$ for $t = 1, 2, \dots, T$.
7. Draw $W_{1:T}^{(k+1)}$ from $p(W_{1:T}|V_{1:T}^{(k+1)}, \gamma_{0:T}, y_{1:T})$.

which is identical to Algorithm 12. So one version of the full CIS algorithm based on $\psi_{0:T}$, $\tilde{\psi}_{0:T}$, $\gamma_{0:T}$, and $\tilde{\gamma}_{0:T}$ is identical to a GIS algorithm. As long as we believe Yu and Meng and don't think the order in which we use each of the DAs in a CIS or GIS algorithm matters much, there doesn't appear to be any benefit to using CIS for DLMs.

3 Application: The Local Level Model

INSERT A PARAGRAPH MOTIVATING WHY WE LOOK AT THE LOCAL LEVEL MODEL

The local level model (LLM) is a DLM with univariate data y_t for $t = 1, 2, \dots, T$ and a univariate latent state θ_t for $t = 0, 1, 2, \dots, T$ that satisfies

$$y_t|\theta_{0:T} \stackrel{ind}{\sim} N(\theta_t, V) \tag{11}$$

$$\theta_t|\theta_{0:t-1} \sim N(\theta_{t-1}, W) \tag{12}$$

with $\theta_0 \sim N(m_0, C_0)$. Here $\theta_t = E[y_t | \theta_{0:T}]$, i.e. the average value of y_t . The states are $\theta_{0:T}$, the scaled disturbances are $\gamma_{0:T}$ with $\gamma_0 = \theta_0$ and $\gamma_t = (\theta_t - \theta_{t-1})/\sqrt{W}$ for $t = 1, 2, \dots, T$, and the scaled errors are $\psi_{0:T}$ with $\psi_0 = \theta_0$ and $\psi_t = (y_t - \theta_t)/\sqrt{V}$ for $t = 1, 2, \dots, T$. The independent inverse Wishart priors on V and W in Section 1 cash out to independent inverse gamma priors for the local level model, i.e. $V \sim IG(\alpha_V, \beta_V)$ and $W \sim IG(\alpha_W, \beta_W)$.

3.1 Base Samplers

The joint density of $(V, W, \theta_{0:T}, y_{1:T})$ is:

$$p(V, W, \theta_{0:T}, y_{1:T}) \propto V^{-(\alpha_V + 1 + T/2)} \exp \left[-\frac{1}{V} \left(\beta_V + \frac{1}{2} \sum_{t=1}^T (y_t - \theta_t)^2 \right) \right] \\ W^{-(\alpha_W + 1 + T/2)} \exp \left[-\frac{1}{W} \left(\beta_W + \frac{1}{2} \sum_{t=1}^T (\theta_t - \theta_{t-1})^2 \right) \right] \exp \left[-\frac{1}{2C_0} (\theta_0 - m_0)^2 \right]$$

This immediately gives the state sampler:

Algorithm 17 (State Sampler for LLM).

1. Draw $\theta_{0:T}$ from $p(\theta_{0:T} | V^{(k)}, W^{(k)}, y_{1:T})$ using FFBS.
2. Draw $(V^{(k+1)}, W^{(k+1)})$ from $p(V, W | \theta_{0:T}, y_{1:T})$.

In step 2, V and W are independent with $V \sim IG(a_V, b_V)$ and $W \sim IG(a_W, b_W)$ where $a_V = \alpha_V + T/2$, $b_V = \beta_V + \sum_{t=1}^T (y_t - \theta_t)^2/2$, $a_W = \alpha_W + T/2$, and $b_W = \beta_W + \sum_{t=1}^T (\theta_t - \theta_{t-1})^2/2$.

The scaled disturbance sampler, the DA algorithm based on the scaled disturbances, is a bit more complicated. In this context $\gamma_0 = \theta_0$ and $\gamma_t = (\theta_t - \theta_{t-1})/\sqrt{W}$ for $t = 1, 2, \dots, T$, and thus $\theta_t = \sqrt{W} \sum_{s=1}^T \gamma_s + \gamma_0$ for $t = 1, 2, \dots, T$. Following (7), we can write the joint posterior of $(V, W, \gamma_{0:T})$ as

$$p(V, W, \gamma_{0:T} | y_{1:T}) \propto V^{-(\alpha_V + 1 + T/2)} \exp \left[-\frac{1}{V} \left(\beta_V + \frac{1}{2} \sum_{t=1}^T (y_t - \gamma_0 - \sqrt{W} \sum_{s=1}^t \gamma_s)^2 \right) \right] \\ \times W^{-(\alpha_W + 1)} \exp \left[-\frac{\beta_W}{W} \right] \exp \left[-\frac{1}{2} \sum_{t=1}^T \gamma_t^2 \right] \exp \left[-\frac{1}{2C_0} (\gamma_0 - m_0)^2 \right] \quad (13)$$

Now V and W are no longer conditionally independent given $\gamma_{0:T}$ and $y_{1:T}$. Instead of attempting the usual DA algorithm, we'll add an extra Gibbs step and draw V and W separately. This gives us the scaled disturbance sampler:

Algorithm 18 (Scaled Disturbance Sampler for LLM).

1. Draw $\gamma_{0:T}$ from $p(\gamma_{0:T} | V^{(k)}, W^{(k)}, y_{1:T})$, possibly using FFBS to sample $\theta_{0:T}$ then transforming.
2. Draw $V^{(k+1)}$ from $p(V | W^{(k)}, \gamma_{0:T}, y_{1:T})$.
3. Draw $W^{(k+1)}$ from $p(W | V^{(k+1)}, \gamma_{0:T}, y_{1:T})$.

In step 2, V is drawn from the same inverse gamma distribution as in step 2 of Algorithm 17. In step 3, the draw of W is more complicated. The density can be written as

$$p(W | V, \gamma_{0:T}, y_{1:T}) \propto W^{-\alpha_W - 1} \exp \left[-\frac{1}{2W} \sum_{t=1}^T \left(y_t - \gamma_0 - \sqrt{W} \sum_{s=1}^t \gamma_s \right)^2 \right].$$

This density isn't any known form and is difficult to sample from. The log density can be written as

$$\log p(W | V, \gamma_{0:T}, y_{1:T}) = -aW + b\sqrt{W} - (\alpha_W + 1) \log W - \beta_W/W + C$$

where C is some constant, $a = \sum_{t=1}^T (\sum_{j=1}^t \gamma_j)^2 / 2V$ and $b = \sum_{t=1}^T (y_t - \gamma_0)(\sum_{j=1}^t \gamma_j) / V$. It can be shown that $b^2 > \frac{32}{9\beta_w}(\alpha_w + 1)^3(1 - 2\text{sgn}(b)/3)$ implies that the density is log concave where

$$\text{sgn}(b) = \begin{cases} 1 & \text{if } b > 0 \\ 0 & \text{if } b = 0 \\ -1 & \text{if } b < 0. \end{cases}$$

This condition is equivalent to $\partial^2 \log p(W|\cdot) / \partial W^2 < 0$ at the W^* that maximizes $\partial \log p(W|\cdot) / \partial W$ and hence guarantees the density is globally log-concave. It turns out that this tends to hold over a wide region of the parameter space — so long as V is smaller or isn't much larger than W . This allows for the use of adaptive rejection sampling in order to sample from this distribution in many cases, e.g. using Gilks and Wild [1992]. An alternative is to use a t approximation to the conditional density as a proposal in a rejection sampler, but this is much more computationally expensive when necessary.

The scaled error sampler is similar to the scaled disturbance sampler, and this is easy to see in the local level model. Here $\psi_0 = \theta_0$ and $\psi_t = (y_t - \theta_t) / \sqrt{V}$ for $t = 1, 2, \dots, T$ so that $\theta_t = y_t - \sqrt{V}\psi_t$ for $t = 1, 2, \dots, T$. From (8) we can write $p(V, W, \psi_{0:T} | y_{1:T})$ as

$$p(V, W, \psi_{0:T}, y_{1:T}) \propto W^{-(\alpha_w + 1 + T/2)} \exp \left[-\frac{1}{W} \left(\beta_w + \frac{1}{2} \sum_{t=1}^T (Ly_t - \sqrt{V}L\psi_t)^2 \right) \right] \\ V^{-(\alpha_v + 1)} \exp \left[-\frac{\beta_V}{V} \right] \exp \left[-\frac{1}{2} \sum_{t=1}^T \psi_t^2 \right] \exp \left[-\frac{1}{2C_0} (\psi_0 - m_0)^2 \right]$$

where we define $Ly_t = y_t - y_{t-1}$ for $t = 2, 3, \dots, T$ & $Ly_1 = y_1 - \psi_0$ and $L\psi_t = \psi_t - \psi_{t-1}$ for $t = 2, 3, \dots, T$ & $L\psi_1 = \psi_1 - 0$. Once again, V and W are no longer conditionally independent given $\psi_{0:T}$ and $y_{1:T}$. In fact, the density seems analogous to (13) with V and W “switching places” so to speak. The scaled error sampler obtained from drawing V and W separately is:

Algorithm 19 (Scaled Error Sampler for LLM).

1. Draw $\psi_{0:T}$ from $p(\psi_{0:T} | V^{(k)}, W^{(k)}, y_{1:T})$, possibly using FFBS to sample $\theta_{0:T}$ then transforming.
2. Draw $V^{(k+1)}$ from $p(V | W^{(k)}, \psi_{0:T}, y_{1:T})$.
3. Draw $W^{(k+1)}$ from $p(W | V^{(k+1)}, \psi_{0:T}, y_{1:T})$.

In step 3, W is drawn from the same inverse gamma distribution as in step 2 of Algorithm 17. Drawing V in step 2 is more complicated, but exactly analogous to drawing W in Algorithm 18. The log density of $V | W, \psi_{0:T}, y_{1:T}$ can be written as

$$\log p(V | W, \psi_{0:T}, y_{1:T}) = -aV + b\sqrt{V} - (\alpha_V + 1) \log W - \beta_V / V + C$$

where again C is some constant, but now $a = \sum_{t=1}^T (L\psi_t)^2 / 2W$ and $b = \sum_{t=1}^T (L\psi_t Ly_t) / W$. So we can use the same methods to sample from this density — adaptive rejection sampling, as in Gilks and Wild [1992], will work as long as $b^2 > \frac{32}{9\beta_V}(\alpha_V + 1)^3(1 - 2\text{sgn}(b)/3)$, and otherwise a t proposal in a rejection sampler will work but will be substantially slower.

3.2 Hybrid Samplers: Interweaving, Alternation and Random Kernel

Section 2.3 contains the details for the interweaving algorithms in the general DLM. In the local level model, there is little to add. We'll consider all four GIS samplers based on any two or three of the base samplers and one CIS sampler. In the GIS samplers, the order of the parameterizations will always be the states ($\theta_{0:T}$), then the scaled disturbances ($\gamma_{0:T}$), then the scaled errors ($\psi_{0:T}$). All of the GIS algorithms and the CIS algorithm are below in Table 1. Note the distributional forms for each of these steps (in some cases a

transformation) are in Section 3.1. In Section 2.3 we saw that one version of the CIS algorithms is the same as the “Error-Dist” GIS algorithm (i.e. the Dist-Error algorithm but flipping the order in which the DAs are used). The CIS algorithm below is not the same as the CIS algorithm which is equivalent to the Error-Dist GIS algorithm, but the difference is only the order in which the DAs are used within each Gibbs step. Thus we don’t expect it to perform much differently from the Dist-Error GIS algorithm, but we include it for completeness.

1. State-Dist GIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\gamma_{0:T}|V, W, \theta_{0:T}] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}]$
2. State-Error GIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\psi_{0:T}|V, W, \theta_{0:T}] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$
3. Dist-Error GIS algorithm:
 $[\gamma_{0:T}|V, W] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}] \rightarrow [\psi_{0:T}|V, W, \gamma_{0:T}] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$
4. Triple GIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\gamma_{0:T}|V, W, \theta_{0:T}] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}] \rightarrow [\psi_{0:T}|V, W, \gamma_{0:T}] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$
5. Full CIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V|W, \theta_{0:T}] \rightarrow [\psi_{0:T}|V, W, \theta_{0:T}] \rightarrow [V|W, \psi_{0:T}] \rightarrow [\theta_{0:T}|V, W] \rightarrow [W|V, \theta_{0:T}] \rightarrow [\gamma_{0:T}|V, W] \rightarrow [W|V, \gamma_{0:T}]$

Table 1: GIS Algorithms for the local level model

Interweaving algorithms are conceptually very similar to alternating algorithms. For every GIS algorithm, there’s a corresponding alternating algorithm where each $[DA_2|V, W, DA_1]$ step is replaced by a $[DA_2|V, W]$ step (here DA_i is a data augmentation for $i = 1, 2$). Table 2 contains each alternating algorithm. Note that there are two possible “hybrid triple” algorithms that we don’t consider here where the move from $\theta_{0:T}$ to $\gamma_{0:T}$ interweaves and while the move from $\gamma_{0:T}$ to $\psi_{0:T}$ alternates and vice versa.

1. State-Dist alternating algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\gamma_{0:T}|V, W] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}]$
2. State-Error alternating GIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\psi_{0:T}|V, W] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$
3. Dist-Error alternating GIS algorithm:
 $[\gamma_{0:T}|V, W] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}] \rightarrow [\psi_{0:T}|V, W] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$
4. Triple alternating GIS algorithm:
 $[\theta_{0:T}|V, W] \rightarrow [V, W|\theta_{0:T}] \rightarrow [\gamma_{0:T}|V, W] \rightarrow [V|W, \gamma_{0:T}] \rightarrow [W|V, \gamma_{0:T}] \rightarrow [\psi_{0:T}|V, W] \rightarrow [V|W, \psi_{0:T}] \rightarrow [W|V, \psi_{0:T}]$

Table 2: Alternating algorithms for the local level model

Finally, we also consider random kernel algorithms. In this context, a random kernel algorithm randomly chooses from the state sampler, scaled disturbance sampler, and scaled error sampler in each iteration where the selection probabilities are constant with respect to the iteration. We consider four random kernel algorithms based on any two or three of the base samplers with an equal probability of selecting each base sampler included in the algorithm. For example, the State-Dist random kernel algorithm selects either the state sampler or the scaled disturbance sampler with equal probability at every iteration, while the triple

random kernel algorithm selections from the state sampler, scaled disturbance sampler, or the scaled error sampler with equal probability at every iteration.

Table 3 contains each algorithm we considered for the local level model. The basic idea here is that the alternating algorithms and the random kernel algorithms should serve as a sort of baseline to compare the corresponding interweaving algorithms against. The GIS algorithm should be slightly faster than the alternating algorithm since the only difference is one step becoming a transformation instead of a random draw, but the difference shouldn't be large. So we'd like the GIS algorithms to have at least as quick mixing as the corresponding alternating algorithms. The random kernel algorithms, however, have to do half as much computation to obtain a single draw (or a third as much computation in the case of the triple random kernel algorithm). Thus in some vague sense, we would like the GIS algorithms to have mixing which is twice as fast as the corresponding random kernel algorithm, or three times as fast in the case of the triple algorithms. We can make this notion precise by considering the effective sample size (ESS) of the Markov chain – we'd like the GIS algorithms to have an ESS about twice as large as the corresponding random kernel algorithms, or three times as large for the triple algorithms.

Type				
Base	State	Scaled Disturbance	Scaled Error	
GIS	State-Dist	State-Error	Dist-Error	Triple
Alt	State-Dist	State-Error	Dist-Error	Triple
RandKern	$\frac{1}{2}\text{State} + \frac{1}{2}\text{Dist}$	$\frac{1}{2}\text{State} + \frac{1}{2}\text{Error}$	$\frac{1}{2}\text{Dist} + \frac{1}{2}\text{Error}$	$\frac{1}{3}\text{State} + \frac{1}{3}\text{Dist} + \frac{1}{3}\text{Error}$
CIS	State-Error for $V W$; State-Dist for $W V$			

Table 3: Each algorithm considered for the local level model

3.3 Simulation Setup

In order to test these algorithms, we simulated a fake dataset from the local level model for various choices of V , W , and T . We created a grid over V - W space with (V, W) ranging from $(10^{-2}, 10^{-2})$ to $(10^2, 10^2)$ and we simulated a dataset for all possible combinations of V and W with each of $T = 10, 100, 1000$. Then for each dataset, we fit the local level model using each algorithm in Table 3. We used the same priors for each model: $\theta_0 \sim N(0, 10^7)$, $V \sim IG(5, 4\tilde{V})$, and $W \sim IG(5, 4\tilde{W})$, mutually independent where (\tilde{V}, \tilde{W}) are the true values of V and W used to simulate the time series. This ensures all priors and thus the posterior are proper and results in independent inverse gamma priors on V and W with prior means equal to the respective true values and moderately large prior variances. We chose these priors for (V, W) for each sampler to sidestep issues of posterior propriety or noninformativeness for variances since they are largely orthogonal to our goals.

For each model and each sampler, we $n = 3000$ draws and threw away the first 500 as burn in. The chains were started at the true values used to simulated the time series, so we can examine the behavior of the chains to determine how well they mix but not how quickly they convergence. Define the effective sample proportion (ESP) for a scalar component of the chain as the effective sample size (ESS, REFERENCE HERE) of the component divided by the actual sample size, i.e. $ESP = ESS/n$. An $ESP = 1$ indicates that the Markov chain is behaving as if it obtains iid draws from the posterior. It's possible to obtain $ESP > 1$ if the draws are negatively correlated and this happens occasionally with some of our samplers, but we round this down to $ESP = 1$ in order to simplify our plots.

3.4 Base Results

Figure 1 contains plots of ESP for V and W in each chain of each base sampler for each of $T = 10$, $T = 100$, and $T = 1000$. We'll focus on $T = 10$ first. The state sampler has a low ESP for V and a high ESP for W when the signal-to-noise ratio, W/V , is larger than one. When the signal-to-noise ratio is smaller than one, on the other hand, the state sampler has a low ESP for W and a high ESP for V . In the usual case where

the signal to noise ratio isn't too different from one, the state sampler has a modest to low ESP for both V and W . Note that the particular values of V and W don't seem to matter at all — just their relative values, i.e. the signal-to-noise ratio W/V . Moving up any diagonal on the plots for V and W in the state sampler, W/V is constant and the ESS appears roughly constant. The basic lesson here is that the state sampler has mixing issues for whichever of V or W is smaller.

Figure 1 tells a different story for the scaled disturbance sampler. When the signal-to-noise ratio is less than one, ESPs for both V and W are nearly 1, i.e. the effective sample size is nearly the actual sample size of the chain. When the signal-to-noise ratio is greater than one, however, ESP for both V and W becomes small, especially for V . Once again the absolute values of V and W don't matter for this behavior — just the relative values. The scaled error sampler has essentially the opposite properties. When W/V is large, it has a near 1 ESP for both V and W . On the other hand, when W/V is small it has a low ESP for both V and W , especially for V . The lesson here seems to be that the scaled disturbances ($\gamma_{0:T}$) are the preferred data augmentation for low signal-to-noise ratios and the scaled errors ($\psi_{0:T}$) are the preferred data augmentation for high signal-to-noise ratios, while the states ($\theta_{0:T}$) are preferred for signal-to-noise ratios near 1.

The plots for $T = 100$ and $T = 1000$ in Figure 1 tell basically the same story, with a twist. Increasing the length of the time series seems to exacerbate all problems without changing the basic conclusions. As T increases, W/V has to be smaller and smaller for the scaled disturbance sampler to have decent mixing, and similarly W/V has to be larger and larger for the scaled error sampler to have decent mixing. Interestingly, the scaled error sampler appears to mix well for both V and W over a larger region of the space $W/V < 1$ than the scaled disturbance sampler does over $W/V > 1$. The state sampler is stuck between a rock and a hard place, so to speak, since as T increases, good mixing for V requires W/V to be smaller and smaller, but good mixing for W requires W/V to be larger and larger.

It's also worth noting that both the scaled error and scaled disturbance samplers run into trouble with their adaptive rejection sampling step in precisely the same region of the parameter space where they have good mixing for both V and W , though as T increases, this only happens in the increasingly extreme ends of the parameter space. More precisely, when $W/V > 1$, $p(W|V, \psi_{0:T}, y_{1:T})$ will often fail to be log concave, and when $W/V < 1$, $p(V|W, \gamma_{0:T}, y_{1:T})$ will often fail to be log concave, but as T increases the degree to which W/V must differ from one (in the appropriate direction) in order for log concavity to often or even occasionally fail increases. Outside of these respective regions, log-concavity of the relevant density failing is an extremely unlikely occurrence. As a result, the adaptive rejection sampling algorithm of Gilks and Wild [1992] won't work in general. Another option is to give up directly sampling from either conditional density and use a metropolis step, perhaps for (V, W) jointly, though we haven't tried that. In general, the sampling algorithm should be prepared to use something other than adaptive rejection sampling if necessary because it's possible that the chain enters a region of the parameter space where the relevant density is not log concave, no matter what the likely values of V and W are. *NOTE: ADD DETAILS ABOUT PRECISELY HOW LARGE OR SMALL W/V HAS TO BE TO THIS PARAGRAPH*

Based on the intuition in Section 2 above, the GIS algorithms should work best when at least one of the underlying base algorithms has a high ESP — the basic idea is that when least one of the underlying algorithms has low autocorrelation, we should have low autocorrelation in the GIS algorithm using multiple DAs. This suggests that the dist-error GIS algorithm will have the best performance of the two DA GIS algorithms for both V and W , especially for W/V far away from one, though when W/V is near one it may offer no improvement, especially for large T . The state-dist GIS algorithm should have trouble with V when W/V is high since both the state sampler and the scaled disturbance sampler have trouble with V when W/V is high. Similarly, the state-error GIS algorithm should have trouble with W when W/V is low since both underlying samplers have trouble with W when W/V is low. Since the triple GIS algorithm adds the state sampler into the dist-error GIS algorithm, it seems plausible that it might improve mixing for one of V or W since for V/W different from one, the state sampler has good mixing for at least one of V or W . The full CIS algorithm, on the other hand, is unlikely to be better than the dist-error GIS algorithm since in a certain sense one algorithm is the same as the other, just with the steps reordered.

We can verify most of these intuitions in Figure 2. First, the state-dist GIS algorithm has high ESP for W except for a narrow band where W/V is near one, though this band becomes much wider as T increases.

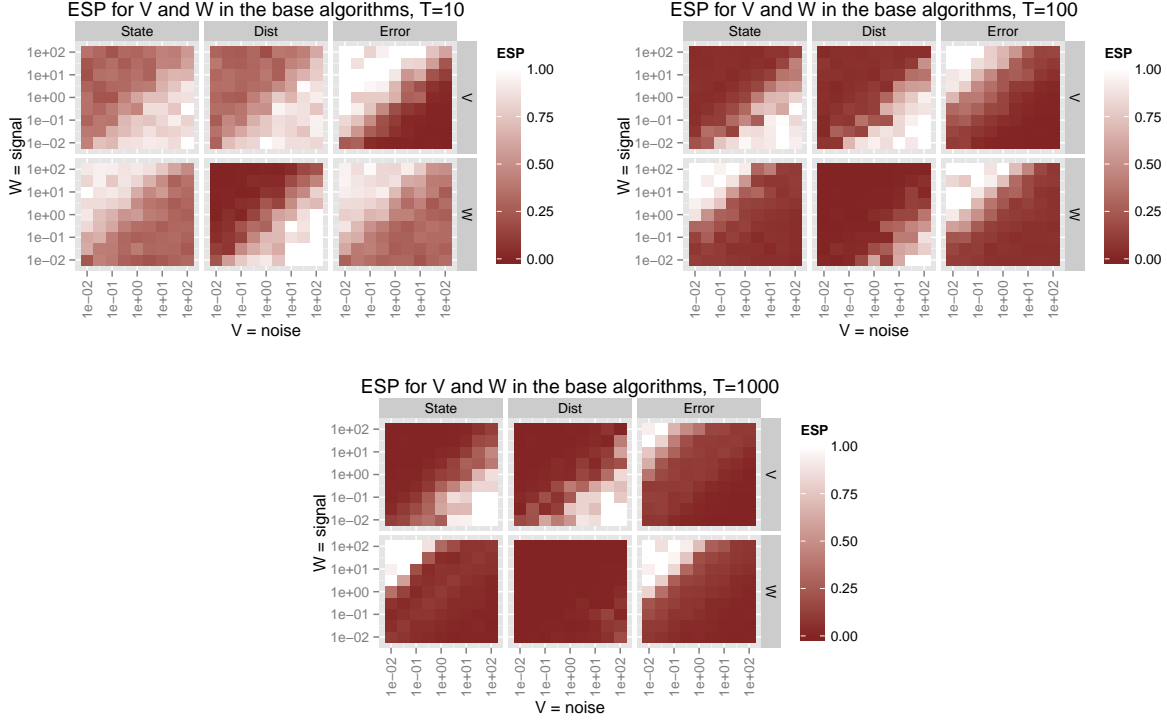


Figure 1: Effective sample proportion in the posterior sampler for a time series of lengths $T = 10$, $T = 100$, and $T = 1000$, for V and W , and for the state, scaled disturbance, and scaled error samplers. X and Y axes indicate the true values of V and W respectively for the simulated data. Note that the signal-to-noise ratio is constant moving up any diagonal. In the upper left the signal is high, in the lower right the noise is high. Note that for plotting purposes, effective sample proportions larger than 1 were rounded down to 1

The state-dist GIS algorithm’s mixing behavior for V appears identical to the original state sampler — high ESP when $W/V < 1$ and poor ESP when $W/V > 1$, and again the good region shrinks as T increases. So this algorithm behaves as expected — it takes advantage of the fact that the state and scaled disturbance DA algorithms make up a “beauty and the beast” pair for W and thus improves mixing for W . However, the two underlying DA algorithms behave essentially identically for V so there is no improvement. Similarly the state-error GIS algorithm’s ESP for W is essentially identical to the state and scaled error algorithms’ ESP for W — high when W/V large and low when W/V small — but for V , the state-error algorithm has a high ESP when W/V isn’t too close to one, especially when T is small. The dist-error GIS algorithm also behaves as predicted — when W/V is not too close to one it has high ESP for both V and W , though as T increases W/V has to be farther away from one in order for the ESPs to be high. The dist-error GIS algorithm behaves apparently identically to the full CIS and triple GIS algorithms, with some differences when T is small. The first of these is not surprising — based on the intuition that the dist-error GIS and full CIS algorithms are the same up to a reordering of each of their steps, we didn’t expect much of a difference. However, we had some hope that the triple GIS algorithm would improve upon the dist-error GIS algorithm somewhat by further breaking the correlation between iterations in the Markov chain. This didn’t happen, and furthermore the state-dist and state-error samplers didn’t improve the ESP for V or W respectively. When the two underlying DA algorithms form a “beast and the beast” pair, there doesn’t seem to be any improvement from interweaving.

Finally Figure 3 allows us to compare the GIS algorithms to the alternating and random kernel algorithms. Note that for the purposes of making a direct comparison, these plots show $ESP/2$ for the three two-

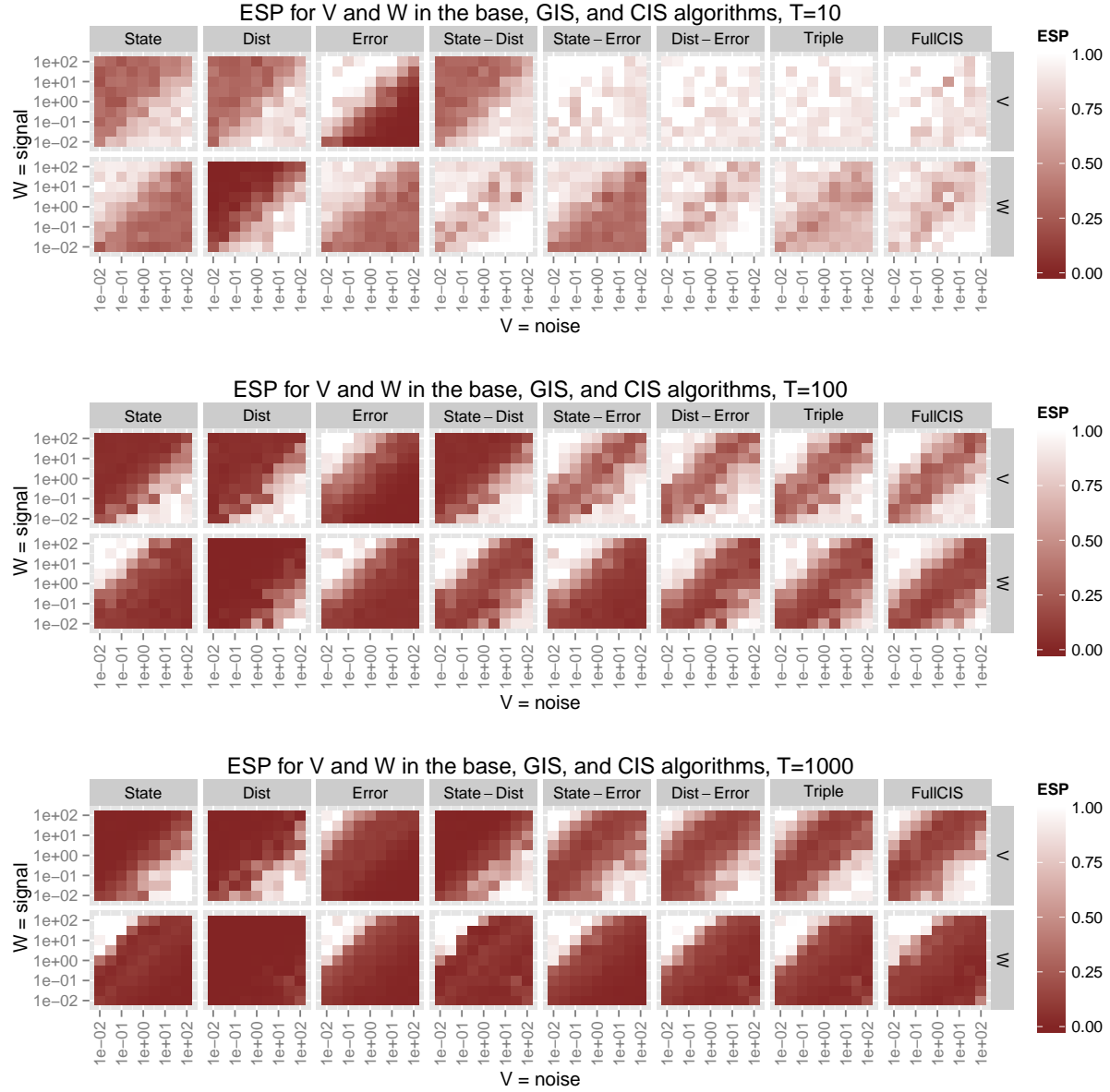


Figure 2: Effective sample proportion in the posterior sampler for V and W in for $T = 10$, $T = 100$, and $T = 1000$, in the state, scaled disturbance and scaled error samplers and for all three GIS samplers based on any two of these. Horizontal and vertical axes indicate the true values of V and W respectively for the simulated data. Note that the signal-to-noise ratio is constant moving up any diagonal. In the upper left the signal is high, in the lower right the noise is high. Note that for plotting purposes, effective sample proportions larger than one were rounded down to one.

DA random kernel algorithms and ESP/3 for the triple random kernel algorithm. We do this because the alternating and itnerweaving algorithms each have to two roughly twice as much computation as the random kernel algorithm in order to complete one full iteration of the sampler, or in the case of the triple algorithms three times as much. The main takeaway is that there doesn't appear to be any difference between itnerweaving and alternating, and the differences between the random kernel and the former two

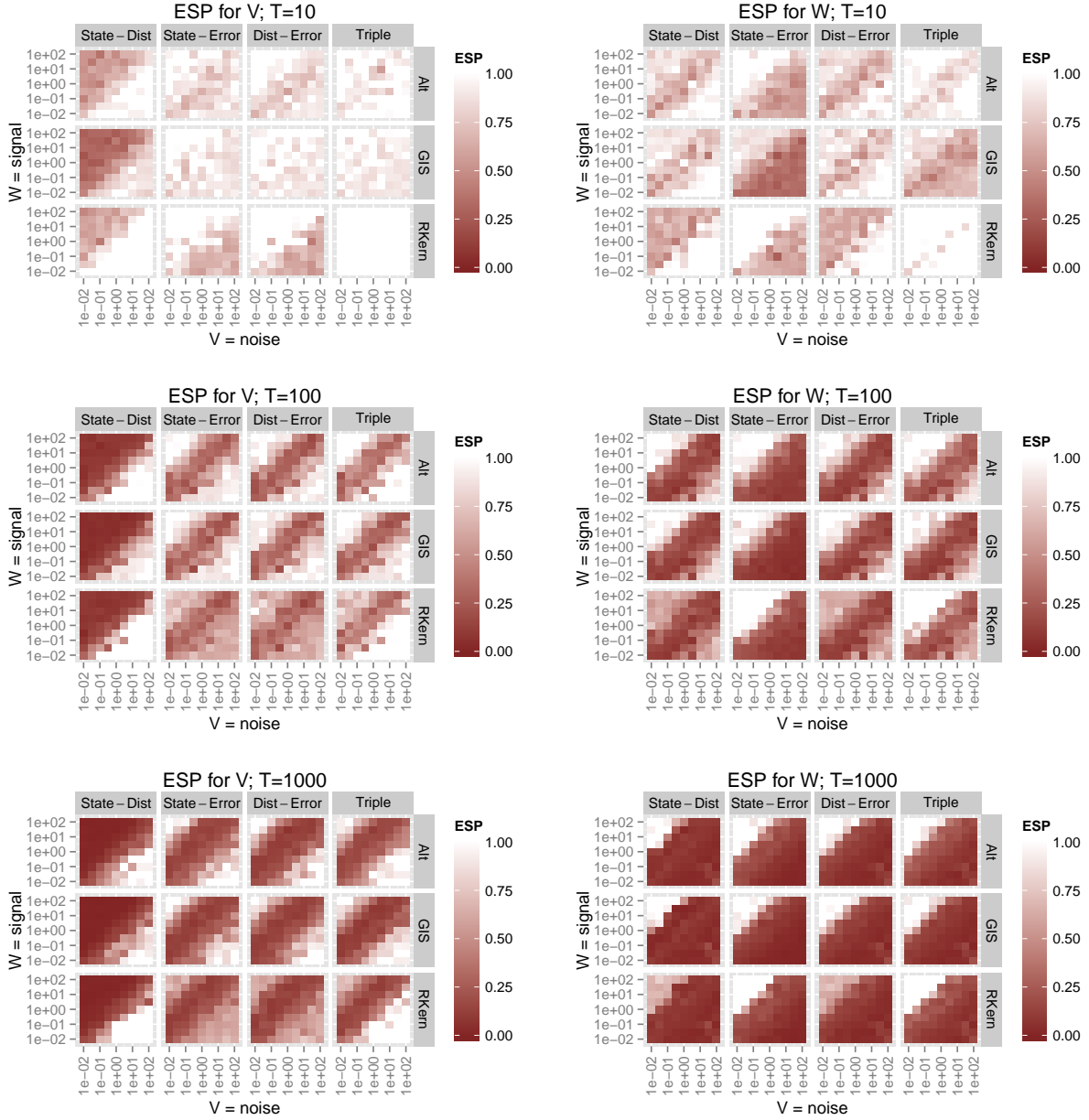


Figure 3: Effective sample proportion in the posterior sampler for a time series of length $T = 10$, $T = 100$, and $T = 1000$, for V and W , and for the GIS and alternating samplers based on the state, scaled disturbance, and scaled error samplers. X and Y axes indicate the true values of V and W respectively for the simulated data. The signal-to-noise ratio is constant moving up any diagonal. In the upper left the signal is high, in the lower right the noise is high. Note that for plotting purposes, effective sample proportions larger than 1 were rounded down to 1. Also note that the *ESP* for the random kernel samplers has been multiplied by 2 or, in the case of the triple kern sampler, by 3, in order to make them comparable to the GIS and alternating samplers.

algorithms are small. For large T , the random kernel algorithm tends to be a bit worse than the GIS and

alternating algorithms in the “good” region of the parameter space, but in the “bad” region the differences aren’t meaningful.

References

- Chris K Carter and Robert Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.
- Sylvia Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, 1994.
- Sylvia Frühwirth-Schnatter. Efficient Bayesian parameter estimation for state space models based on reparameterizations. *State Space and Unobserved Component Models: Theory and Applications*, pages 123–151, 2004.
- Walter R Gilks and Pascal Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, pages 337–348, 1992.
- James P Hobert and Dobrin Marchev. A theoretical comparison of the data augmentation, marginal augmentation and PX-DA algorithms. *The Annals of Statistics*, 36(2):532–554, 2008.
- Siem Jan Koopman. Disturbance smoother for state space models. *Biometrika*, 80(1):117–126, 1993.
- Jun S Liu and Ying Nian Wu. Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274, 1999.
- Xiao-Li Meng and David A Van Dyk. Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86(2):301–320, 1999.
- Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.
- Giovanni Petris, Patrizia Campagnoli, and Sonia Petrone. *Dynamic linear models with R*. Springer, 2009.
- David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1), 2001.
- Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question - an ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011.