Practical Issues with State-Space Models with Mixed Stationary and Non-Stationary Dynamics Technical Paper No. 2010-1

Thomas A. Doan [*]

January 2010

Abstract

State-space models, and the state-space representation of data, are an important tool for econometric modeling and computation. However, when applied to observed (rather than detrended) data, many such models have a mixture of stationary and non-stationary roots. While Koopman (1997) and Durbin and Koopman (2002) provide "exact" calculations for models with non-stationary roots, these have not yet been implemented in most software. Also, neither the Koopman article nor the Durbin and Koopman book address (directly) the handling of models where a unit root is shared among several series—a frequent occurrence in state-space models derived from DSGE's. This paper provides a unified framework for computing the finite and "infinite" components of the initial state variance matrix which is both flexible enough to handle mixed roots and also faster (even for purely stationary models) than standard methods. In addition, it examines some special problems that arise when the number of unit roots is unknown *a priori*.

Keywords: Kalman filter, state space methods, non-stationary model

[*] Estima, 1560 Sherman Ave #510, Evanston, IL 60201 USA, email:tomd@estima.com

The author thanks participants in seminars at the Bank of England and Cass Business School for helpful comments, and Tara Sinclair and Ricardo Mestre for providing code and data for their papers. Comments are welcome.

1 Introduction

State-space models, and the state-space representation of data, are an important tool for econometric modeling and computation. However, when applied to observed (rather than detrended) data, many such models have a mixture of stationary and non-stationary roots. While Koopman (1997) and Durbin and Koopman (2002) provide "exact" calculations for models with non-stationary roots, these have not yet been implemented in most software. Also, neither the Koopman article nor the Durbin and Koopman book address (directly) the handling of models where a unit root is shared among several series—a frequent occurrence in state-space models derived from DSGE's.

This paper provides a unified framework for computing the finite and "infinite" components of the initial state variance matrix which is both flexible enough to handle mixed roots and also faster (even for purely stationary models) than standard methods. This is based upon a method for solving for the pre-sample variance which is known in the engineering literature, but largely unknown within economics. This is extended to handle non-stationary roots. In addition, it examines some special problems that arise when the number of unit roots is unknown *a priori*.

The paper is organized as follows. In section 2, we introduce the notation to be used and describe some common small state-space models, paying particular attention to the types of roots. Section 3 works through a step of the Kalman filter with a diffuse prior, showing the differences between an exact treatment of the initial conditions and the commonly used approximate handling. Section 4 describes other algorithms for computing the initial variances for fully stationary models, while section 5 describes a more efficient method based upon the Schur decomposition of **A**. Section 6 extends this to non-stationary models. Section 7 offer two examples from the literature, highlighting the difference in calculation methods. Section 8 concludes.

2 State-Space Models

We will use the following as our general description of a state-space model:

$$\mathbf{X}_t = \mathbf{A}_t \mathbf{X}_{t-1} + \mathbf{Z}_t + \mathbf{F}_t \mathbf{w}_t \tag{1}$$

$$\mathbf{Y}_t = \mu_t + \mathbf{c}_t' \mathbf{X}_t + \mathbf{v}_t \tag{2}$$

$\{\mathbf{w}_t, \mathbf{v}_t\}$ is assumed to be serially independent. Our main focus will be on the properties of (1), and, in particular, that equation with time-invariant component matrices, $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{F}$ and $E\mathbf{w}_t\mathbf{w}_\mathbf{t}'$.

The execution of the Kalman filter requires a pre-sample mean and variance. The ergodic variance is the solution Σ_0 to:

$$\mathbf{A}\Sigma_0\mathbf{A}' + \Sigma_\mathbf{w} = \Sigma_0 \tag{3}$$

where $\Sigma_\mathbf{w} = \mathbf{F}\left(E\mathbf{w}_t\mathbf{w}_t'\right)\mathbf{F}'$. Where this exists, it's the ideal starting variance since it is the one initializer that is consistent with the structure of the model. However, it's well known that there is no solution to (3) if $\mathbf{A}$ has roots on the unit circle. In practice, the most common way to handle this is use an approximate "diffuse" prior, that is, a pre-sample mean of 0 with a covariance matrix which is a diagonal matrix with "large" numbers. While not completely unreasonable, there are a number of numerical problems with this approach, particularly if the estimates of state variances are of interest.

In the Unobserved Components (UC) models analyzed extensively in Harvey (1989), the $\mathbf{A}$ matrices have all unit roots: the local level model has a single unit root, the local trend has a double (repeated) unit root, and the seasonal models have $S-1$ seasonal roots of 1 (other than 1 itself). The state-space representation applied to ARMA models, on the other hand, has all roots inside the unit circle in order to permit the calculation of the full sample likelihood—the non-stationarity of the ARIMA model is typically handled by differencing the data and applying the state-space techniques to that.

A model with a mix of roots can be constructed by "adding" independent components. When this is done, the $\mathbf{A}$ matrix is formed by concatenating the component $\mathbf{A}$'s diagonally. For instance, if the observable is the sum of a local trend plus a AR(2) cycle, the $\mathbf{A}$ matrix is:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \phi_1 & \phi_2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

This has a repeated unit root and two stable roots (assuming the AR(2) process is stationary).

Koopman (1997) and Durbin and Koopman (2002) provide a (relatively) simple exact method for handling diffuse priors in models with (any type of) unit roots.[1]

[1] For the Kalman filtering calculations needed for calculating the likelihood, this is only slightly

The Durbin-Koopman textbook also describes a way of handling a mixture of unit and stationary roots, but offers little guidance as to how to handle this in any situation where the states don't partition nicely as in (4). For instance, their description of the non-stationary ARIMA model uses $y_t, \Delta y_t, \Delta y_{t-1}, ...$ as the states rather than $y_t, y_{t-1}, y_{t-2}, ...$ However, this type of transformation isn't always practical—in some cases, the state-space model is derived from a more complicated problem, and the precise positioning of states may not be well known to the user.[2]

The method proposed in this paper uses the properties of the $\mathbf{A}$ and $\Sigma_{\mathbf{W}}$ matrices only (thus not requiring any ingenuity on the part of the user) to derive a partially diffuse, partially stationary initialization.

3 Diffuse Prior

In practical state-space models, it's rare for all the eigenvalues of the transition matrix to be inside the unit circle. In the simplest of the non-stationary models, the local level model, the "solution" for (3) is $\mathrm{var}(\mathbf{X}) = \infty$. Suppose that we use a prior of mean 0 and variance κ where κ is "large". The Kalman updating equation for the mean is

$$x_{1|1} = x_{1|0} + \frac{\sigma^2_{1|0}}{\sigma^2_{1|0} + \sigma^2_v}\left(y_1 - x_{1|0}\right)$$

With $\sigma^2_{1|0} = \kappa + \sigma^2_w$, if κ is large relative to σ^2_w and σ^2_v this will be approximated by

$$x_{1|1} = x_{1|0} + 1.0\left(y_1 - x_{1|0}\right)$$

or $x_{1|1} = y_1$. This shouldn't come as a surprise; with a very high prior variance, the data will dominate completely. The only data information after one data point is that one value. This calculation isn't particularly sensitive to the precise value of κ as long as it's big enough.

The variance calculation is more problematical. The calculation (in practice) is done by:

$$\sigma^2_{1|1} = \sigma^2_{1|0} - \frac{\left(\sigma^2_{1|0}\right)^2}{\sigma^2_{1|0} + \sigma^2_v}$$

While we can rearrange this (algebraically) to give $\sigma^2_{1|1} \approx \sigma^2_v$, the calculation will be done in software as the subtraction above. Subtracting two very large and

more complicated than the standard Kalman filter. Kalman smoothing isn't quite as straightforward, and is, in fact, described incorrectly in the Koopman article.

[2] For instance, to produce the state-space representation for a model with expectational terms, the state vector can require augmentation with lags, with the number depending upon the precise form of each equation in the model.

almost equal numbers to produce a small number runs the danger of a complete loss of precision. With the standard 15 digit precision on a modern computer, if κ is greater than $10^{15}\sigma_v^2$, the calculation above will give zero.

So in order for this to give us the desired results by using large values of κ, we need to choose a value which is large enough, but not too large. For this model, we can probably figure out a way to do that. With a more complicated model with multiple variances, some of which might be quite small, we're less likely to find such a value.

Note, by the way, that one suggested strategy is to try several values for κ, and check for sensitivity.[3] While better than just picking an arbitrary number, looking for sensitivity on estimates of the mean will likely overlook the more troublesome calculation of the variance. And the calculation that is *most* affected by approximation error is the state variance for Kalman *smoothing*. Those are highly suspect in any calculation done this way.[4]

This calculation is effectively identical if the model is, in fact, stationary, but a diffuse prior is being used for convenience. If, instead of the random walk, the underlying state equation is $x_t = \rho x_{t-1} + w_t$ with $|\rho| < 1$, then we still will get $x_{1|1} \approx y_1$ and $\sigma_{1|1} \approx \sigma_v^2$. This is not the same as what we would get if we used the ergodic variance, but isn't necessarily an unreasonable procedure.[5]

An alternative to approximating this was provided by Koopman (1997), which is *exact diffuse initialization*. This was later refined in Durbin and Koopman (2002). This is done by doing the actual limits as $\kappa \to \infty$. It is implemented by writing covariance matrices in the form $\Sigma_\infty \kappa + \Sigma_*$ where Σ_∞ is the "infinite" part and Σ_* is the "finite" part. These are updated separately as needed.

This works in the following way for the local level model. The prior is mean 0. The prior covariance is $[1]\kappa + [0]$.[6] Moving to $\sigma_{1|0}$ adds the finite σ_w^2, producing $[1]\kappa + [\sigma_w^2]$. The predictive variance for y_1 further adds σ_v^2, to make $[1]\kappa + [\sigma_w^2 + \sigma_v^2]$. The Kalman gain is the multiplier on the prediction error used in updating the state. That's

$$\left([1]\kappa + [\sigma_w^2]\right)\left([1]\kappa + [\sigma_w^2 + \sigma_v^2]\right)^{-1} \tag{5}$$

Inverses can be computed by matching terms in a power series expansion in κ^{-1}

[3] For instance, Koopman, Shepard, and Doornik (1999) suggest calculating once with an standard "large" value, then recalculating with a revised value based upon the initial results.

[4] In a model with more than one diffuse state, the filtered covariance matrix after a small number of data points will still have some very large values. Those are only reduced to "finite" numbers as the result of the smoothing calculation, which uses information obtained by calculating out to the end of the data set and back, accumulating roundoff error along the way.

[5] With the ergodic variance, $x_{1|1}$ will be a weighted average of 0 and y_1 with weights equal to the ergodic variance and σ_v^2 respectively.

[6] We'll keep the different components in brackets.

as shown in Appendix A. For computing the mean, all we need are the 1 and κ^{-1} terms, which allows us to write (5) as approximately:

$$\left([1]\kappa + [\sigma_w^2]\right)\left([0] + [1]\kappa^{-1}\right) = [1] + [\sigma_w^2]\kappa^{-1}$$

where the approximation in the inverse will produce only an additional term on the order of κ^{-2}. We can *now* pass to the limit (that is, ignore all but the finite term) and get the Kalman gain as 1 (exactly), as we expected.

Getting the filtered variance correct requires greater accuracy. For that, we'll need to expand the inverse to the κ^{-2} level. We need this because the updating equation is:

$$\left([1]\kappa + [\sigma_w^2]\right) - \left([1]\kappa + [\sigma_w^2]\right)\left([1]\kappa + [\sigma_w^2 + \sigma_v^2]\right)^{-1}\left([1]\kappa + [\sigma_w^2]\right)$$

and the inverse is sandwiched between two factors, each with a κ. As a result, terms in κ^{-2} in the inverse will produce a finite value when this is multiplied out, and only κ^{-3} and above will be negligible. The second order expansion of the required inverse is

$$\left([0] + [1]\kappa^{-1} - [\sigma_w^2 + \sigma_v^2]\kappa^{-2}\right)$$

The calculation of the filtered variance is most conveniently done as

$$\left(1 - \left([1]\kappa + [\sigma_w^2]\right)\left([1]\kappa + [\sigma_w^2 + \sigma_v^2]\right)^{-1}\right)\left([1]\kappa + [\sigma_w^2]\right) \tag{6}$$

With the second order expansion, the product

$$\left([1]\kappa + [\sigma_w^2]\right)\left([1]\kappa + [\sigma_w^2 + \sigma_v^2]\right)^{-1}$$

produces

$$1 - \sigma_v^2\kappa^{-1} - O(\kappa^{-2}) \tag{7}$$

Plugging that into (6) results in $\sigma_v^2 + O(\kappa^{-1})$. Passing to the limit gives us the result we want.

Note that the calculation here that causes the problem for large, but finite values, is subtracting (7) from 1. If we have a loss of precision there, we won't get σ_v^2 out; we'll get zero.

While we started with an "infinite" variance, after seeing one data point, we now have $x_{1|1} = y_1$, $\sigma_{1|1} = \sigma_v^2$, which are perfectly reasonable finite values. We can Kalman filter in the standard fashion from there.

4 Other Algorithms for Calculating the Ergodic Variance

While quadratic in $\mathbf{A}$, (3) is linear in the elements of Σ_0. The standard textbook solution for this[7] is to rearrange it into the linear system:

$$[\mathbf{I} - \mathbf{A} \otimes \mathbf{A}] vec(\Sigma_0) = vec(\Sigma_{\mathbf{W}}) \tag{8}$$

As written, this has some redundant elements, since Σ_0 is symmetric. Still, with those eliminated, it requires solving a linear system with $n(n+1)/2$ components. Since solution of a system of equations is $O(N^3)$ in arithmetic operations, this makes this solution procedure $O(n^6)$. This starts to dominate the calculation time for even fairly modest values of n.[8]

There are some special cases where the calculation of (3) can be simplified considerably. For instance, if the $\mathbf{A}$ matrix takes a form such as:

$$\begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \cdots & \varphi_n \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

which is the transition matrix for an AR(n) process, the bottom $(n-1) \times (n-1)$ corner of $\mathbf{A}\Sigma_0\mathbf{A}'$ is just the top $(n-1) \times (n-1)$ corner of Σ_0. Thus, given solutions for $\sigma_{1j}^{(0)}$, the remainder of the matrix can be filled in by solving recursively $\sigma_{i+1,j+1}^{(0)} = \sigma_{i,j}^{(0)} + \sigma_{i+1,j+1}^{(w)}$. So the problem reduces to a linear system with just the n unknowns $\sigma_{1j}^{(0)}$. The solution to this is $O(n^3)$.

An analogous simplification can be done with the block form of (9) that occurs when the model is a vector autoregression. With m variables and p lags, a direct solution of (8) with $n = mp$ total states becomes prohibitive with all but very small models. The recursive solution cuts this down to solving an mp^2 system of equations, which knocks a factor of m^2 out of the number of calculations compared with the textbook solution. With a large value of p, however, this will still be quite expensive.

In Johansen (2002), the author needed a better algorithm for solving (3) for

[7] See, for instance, Hamilton (1994), page 378 or Durbin and Koopman (2002), page 112.

[8] The most time-consuming calculation in one Kalman filter update is computing $\mathbf{A}\Sigma_{t|t}\mathbf{A}'$ which is $O(n^3)$, repeated T times. Thus, if $n^3 \gg T$, $n^6 \gg n^3T$ and the greatest time will be spent on the initial variance calculation.

precisely this reason—the corrections needed the ergodic solution for a state-space representation for a VAR. His proposal was to transform (3) by taking a matrix $\mathbf{P}$ such that $\mathbf{PAP}^{-1} = \Lambda$ and $\mathbf{P}\Sigma_w\mathbf{P}^{-1} = \mathbf{D}$, where Λ and $\mathbf{D}$ are diagonal. Pre-multiplying (3) by $\mathbf{P}$ and post-multiplying by $\mathbf{P}^{-1}$ and inserting $\mathbf{P}^{-1}\mathbf{P}$ in two places produces the equation:

$$\mathbf{PAP}^{-1}\mathbf{P}\Sigma_0\mathbf{P}^{-1}\mathbf{PA}'\mathbf{P}^{-1} + \mathbf{P}\Sigma_\mathbf{w}\mathbf{P}^{-1} = \mathbf{P}\Sigma_0\mathbf{P}^{-1} \tag{10}$$

Defining $\Sigma_p = \mathbf{P}\Sigma_0\mathbf{P}^{-1}$ and substituting the reductions for $\mathbf{A}$ and $\Sigma_\mathbf{w}$ gives us:

$$\Lambda\Sigma_p\Lambda + \mathbf{D} = \Sigma_p$$

Because Λ is diagonal, this reduces to the set of equations:

$$\sigma_{ij}^{(p)}\lambda_i\lambda_j = d_{ij} + \sigma_{ij}^{(p)} \tag{11}$$

Thus, we can directly compute $\sigma_{ij}^{(p)}$, which will also be a diagonal matrix (since $\mathbf{D}$ is diagonal) and then can recover the desired matrix with $\Sigma_0 = \mathbf{P}^{-1}\Sigma_p\mathbf{P}$. Because finding eigenvalues and vectors of an $n \times n$ matrix is an $O(n^3)$ calculation, this reduces the burden by a factor of $O(n^3)$.[9]

One thing to note is that the requirement that $\mathbf{P}$ be a *joint* diagonalizing matrix isn't really necessary. The solution for (11) is very simple whether $\mathbf{P}\Sigma_w\mathbf{P}^{-1} = \mathbf{D}$ is diagonal or not. One complication with this (depending upon the software used) is that, in general, $\mathbf{P}$ and Λ will be complex matrices, and Σ_p will be Hermitian rather than symmetric. Allowing for complex roots, the λ_j in (11) needs to be conjugated.

A more serious problem is that not all state matrices will be diagonalizable. Although it's a case with unit roots (which will be covered later), the standard local trend model transition matrix

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

isn't diagonalizable, and, in fact, is already in its Jordan form. State-space models solved out of DSGE's also frequently have repeated eigenvalues.

[9] Because the final step in computing eigenvalues is iterative, the computational complexity isn't precisely known. The $O(n^3)$ is based upon a fixed number of iterations per eigenvalue.

5 Schur Decomposition Applied to Stationary Models

A result from the engineering literature (Kitagawa (1977)) seems to have been largely ignored in the economics literature. (3) is known as the Lyapunov equation. Kitagawa uses a Schur decomposition of the $\mathbf{A}$ matrix to reduce the calculation time for solving this to $O(n^3)$, the same as the eigen decomposition. Because the Schur decomposition always exists, this can be employed in situations where the eigen calculation can't. Since the Schur decomposition is quite a bit less ambitious than the eigen decomposition, this actually is quicker when both can be applied.[10]

We provide here a description of Kitagawa's technique, in part because much of that paper is spent defininining the Schur decomposition, and in part because it has several typos.

Every square matrix $\mathbf{A}$ has a (not necessarily unique) representation as $\mathbf{A} = \mathbf{QRQ}^{-1}$ where $\mathbf{Q}$ is a unitary matrix[11] and $\mathbf{R}$ is an upper (or "right") triangular matrix. This is a Schur decomposition of $\mathbf{A}$. This will likely produce complex matrices; to avoid that, it's possible to use the "real" Schur decomposition, which also always exists. For that, $\mathbf{Q}$ is a (real) unitary matrix, and $\mathbf{R}$ is a real matrix that is block-upper triangular, where the diagonal blocks are either 1×1 (for real eigenvalues) or 2×2 for pairs of complex eigenvalues.

By the same type of changes made in producing (10) (pre-multiplying (1) by $\mathbf{Q}$ and replacing $\mathbf{X}_{t-1}$ with $\mathbf{Q^{-1}QX}_{t-1}$), we can reduce the solution to:

$$\mathbf{QAQ^{-1}Q}\Sigma_0\mathbf{Q}^{-1}\mathbf{QA'Q^{-1}} + \mathbf{Q}\Sigma_{\mathbf{w}}\mathbf{Q}^{-1} = \mathbf{Q}\Sigma_0\mathbf{Q}^{-1}$$

or

$$\mathbf{R}\Sigma_q\mathbf{R}' + \mathbf{Q}\Sigma_{\mathbf{w}}\mathbf{Q}^{-1} = \Sigma_q \tag{12}$$

We'll describe the calculation based upon the real Schur decomposition. The $\mathbf{R}$ matrix can be blocked in the following way based upon the diagonal blocks, which we number from 1 to p. The $\mathbf{R}_{i,j}$ will be 1×1, 1×2, 2×1 or 2×2 depending upon the sizes of $\mathbf{R}_{i,i}$ and $\mathbf{R}_{j,j}$. (12) can then be written as the set of equations over i, j

[10] The solution of the Lyapunov equation by eigen decomposition is apparently also known in the engineering literature, as it's mentioned in passing by Kitagawa.

[11] The inverse of a unitary matrix is its conjugate transpose, or, for a matrix with only real elements, its simple transpose.

combinations of:

$$\sum_{l=i}^{p} \mathbf{R}_{i,l} \left\{ \sum_{m=j}^{p} \Sigma_{l,m}^{(q)} \left(\mathbf{R}_{j,m}\right)' \right\} + \mathbf{W}_{i,j} = \Sigma_{i,j}^{(q)} \tag{13}$$

which uses the block triangularity of **R** in the lower limits on the two sums.[12] This can be solved recursively starting with $i = j = p$, which is:

$$\mathbf{R}_{p,p}\Sigma_{p,p}^{(q)}\mathbf{R}_{p,p} + \mathbf{W}_{p,p} = \Sigma_{p,p}^{(q)}$$

This takes the form of a Lyapunov equation, but with matrices that are at most 2×2, so the textbook solution method can be employed. Solving (13) at i, j requires $\Sigma_{l,m}^{(*)}$ for all combinations with $m > j$ or $m = j$ and $l > i$. Thus the solution order is to work up column p, then solve for $p-1, p-1$, then work up column $p-1$, etc. The largest system that needs to be solved is a standard linear system with four unknowns for the off-diagonal 2×2.

Done literally, (13) is an $O(n^4)$ calculation, since for each of the $O(n^2)$ combinations i, j the double sum of terms to the right or below is $O(n^2)$. However, the term in braces can be computed just once for each l in column j.[13] Each of those sums is $O(n)$, done n times (over l) and n times (over j), thus $O(n^3)$. However, they produce small matrices[14], so the sum over l is $O(n)$, done $O(n^2)$ times over i, j, again giving the desired $O(n^3)$.

The final step is to transform back to the original space with $\Sigma_0 = \mathbf{Q}'\Sigma_q\mathbf{Q}$, which is, itself, an $O(n^3)$ calculation.

Table 1 gives a comparison of the Schur and eigen methods with the textbook calculation for various numbers of states. These are in number of calculations per second, and show the rapid deterioration of speed with size for the standard calculation.[15]

6 Generalized Ergodic Initialization

The extension to non-stationary models is straightforward. Appendix B shows that the estimated means and variances of the states aren't dependent upon the

[12] Note that $\Sigma_{i,j}^{(q)}$ appears in a term in the double sum, which will have to be pulled out when solving.

[13] When we start on column j, we don't yet know $\Sigma_{l,j}^{(*)}$ for $l \leqslant j$. So the terms involving those are initially omitted and are added into term l once we've solved l, j.

[14] Depending upon the sizes of the blocks, each is a 1×1, 1×2, 2×1 or 2×2 matrix.

[15] These were done on a 2GHz Dell Laptop running Windows Vista. The standard method was solved using LU decomposition.

Table 1: Speed Comparison

n	Schur	Eigen	Standard
10	4464.3	1976.3	847.5
20	961.5	339.0	24.6
30	250.0	101.0	2.5
50	31.3	22.5	.1
100	4.3	3.2	

precise form of the diffuse prior, so we will choose an identity matrix in the **Q** transformed representation.

The solutions for the diagonal blocks in (13) will be in the general form of a Lyapunov equation. If $\mathbf{R}_{j,j}$ has unit eigenvalues, then there is no solution. If the eigenvalues are outside the unit circle, there will generally be a solution, but it is unlikely to be useful in economic applications, so we treat them the same as the unit roots.[16] Classify the diagonal blocks as stationary or non-stationary. Make $\Sigma^{(q)}_{i,j} = 0$ for any block where either i or j is a non-stationary block and otherwise solve (13) as described in the previous section. In a parallel "diffuse matrix", make $\Sigma^{(\infty)}_{i,j} = I$ if $i = j$ and i is a non-stationary block and make it zero otherwise. Note that, if the blocks were ordered so the non-stationary roots were at the lower right, this would have Σ_q blocked with non-zeros only in the top left and Σ_∞ being blocked with an identity matrix in the lower right. Since sorting the eigenvalues in the Schur decomposition is a complicated and time-consuming task, doing the equivalent calculation while leaving them in the order in which they're originally produced is the best strategy.

The finite and infinite components in the original representation are then produced by back transforming by the **Q** matrix as before. The result of this we will call the *Generalized Ergodic Initialization* for the model. Note that it isn't unique since there are many equivalent choices for the diffuse component.

Because of the possibility of round-off error, the test for unit eigenvalues has to allow for some flexibility, that is, the test needs to be whether the roots are bigger than something like .9999999. There are some practical problems which arise when a model has near unit roots, or roots that might, or might not be unit. One of these problems is that there's no continuous definition of the likelihood function when you move from a near unit root to a unit root. The standard recursive definition of the likelihood is based upon the predictive density for the observable. With a large, but finite, variance, that likelihood element is computable, although the

[16] Kitagawa allows for explosive roots, and, in fact, his example includes some.

log likelihood element will be a large negative number. If the predictive variance is infinite, the likelihood is, theoretically, zero and thus the log likelihood isn't defined.

The only way to handle this is to use the conditional rather than unconditional likelihood. That's done automatically in case a unit root is detected. For a state-space model, the conditional likelihood is calculated by not including a certain number of early data points. The number of data points skipped needs to be at least large enough so that the observables cover the possible number of unit roots.[17]

The other problem is that the calculation is also discontinuous due to the shift in algorithm once a root gets inside the zone where we treat it as a unit root. We've found this to be the more vexing of the two problems. The discontinuity itself isn't as significant a problem as the lack of (true) differentiability that it implies. The arc derivatives, which should converge to the computed directional derivatives, don't. As a result, standard procedures for selecting a step size in hill-climbing methods like BFGS or BHHH will fail to work properly.

In order to finesse this, we have found that the simplest solution was to use conditional likelihood with a fully diffuse prior for a certain number of iterations in order to get the estimates relatively close to the proper range. At that point, we switched over to the correctly computed variance. While the fully diffuse prior isn't really correct, it's close enough to use as an improvement on the initial guess values for the parameters.

7 Examples

We now look at two examples where approximate and fully diffuse priors were used and see how the calculations are affected by different ways of handling the initial conditions. These were not chosen as particularly extreme examples—there are just examples with more than one observable, state models with combinations of stationary and non-stationary roots for which the estimating code has been graciously made available, so the precise calculations used can be reproduced.

[17] If there is only one observable, this would mean if there are r potential unit roots, one should condition on r data points. If there is more than one observable, it could be fewer. In most cases, as long as the number of observables times the number of conditioning data points is at least as large as r, the likelihood will be consistently defined.

7.1 Example 1

Sinclair (2009) estimates a bivariate time series model with US log GDP and unemployment rate as the observables. Each series is modeled with a local trend with fixed growth plus an AR(2) cycle.

$$\mathbf{X}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_1^y & \phi_2^y & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \phi_1^u & \phi_2^u \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_{t-1} + \begin{bmatrix} \mu^y \\ 0 \\ 0 \\ \mu^u \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_{1t}^y \\ w_{2t}^y \\ w_{1t}^u \\ w_{2t}^u \end{bmatrix}$$

$$\mathbf{Y}_t = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \mathbf{X}_t$$

This has two unit roots among six states, and has four shocks (there are no measurement errors). We look at her model without the break. Sinclair's handling was to use an approximate diffuse prior with 10^7 diagonal elements, with estimates computed conditioning on four data points to avoid problems with a poorly defined likelihood for those initial values. (With six diffuse states and two equations, this could safely be done conditioning on just three data points.) Table 2 shows the smoothed state and variance estimates at $t = 1$, $t = 4$ and $t = 10$ with the approximate fully diffuse prior, with an approximate diffuse prior with 10^8 rather than 10^7, an exact fully diffuse prior and the mixed ergodic prior.

As we can see from this, the approximate diffuse priors do a good job of computing the smoothed mean at $t = 1$, but the variance calculation is quite sensitive. In this case, 10^7 is already a bit too large—10^6 (not shown) produces a similar result to the exact diffuse prior.

Table 3 compares the maximum likelihood estimates done four ways. The first three use the same conditioning on four data points as in the paper. The first column is the approximate diffuse prior as in the paper, the second is the exact fully diffuse prior and the third uses the mixed ergodic initialization. The final column does the mixed ergodic calculation conditioning on the minimal amount of information. The first five rows have the AR coefficients and the mean growth rate for y. (The mean growth rate of u is pegged at zero). The covariance matrix of the shocks is estimated in Choleski factor form. As we can see, the effect of approximating the diffuse prior is negligible. In this case, the loss of data by excess conditioning seems (comparing third and fourth columns) to be less of a

Table 2: Sinclair Model

	GDP Trend	GDP Cycle	UR Trend	UR Cycle
Time=1				
10^7 Diffuse	739.824	-1.049	3.770	-0.070
	1.761	1.761	0.337	0.337
10^8 Diffuse	739.824	-1.049	3.770	-0.070
	1.768	1.819	0.880	1.008
Exact Diffuse	739.824	-1.049	3.770	-0.070
	1.761	1.761	0.325	0.325
Ergodic	738.727	0.048	3.880	-0.180
	0.365	0.365	0.105	0.105
Time=4				
10^7 Diffuse	738.421	2.915	5.676	-1.876
	0.328	0.328	0.094	0.094
10^8 Diffuse	738.421	2.915	5.676	-1.876
	0.328	0.328	0.094	0.094
Exact Diffuse	738.421	2.915	5.676	-1.876
	0.328	0.328	0.094	0.094
Ergodic	738.520	2.816	5.648	-1.848
	0.301	0.301	0.093	0.093
Time=10				
10^7 Diffuse	748.549	-1.965	4.150	1.450
	0.300	0.300	0.093	0.093
10^8 Diffuse	748.549	-1.965	4.150	1.450
	0.300	0.300	0.093	0.093
Exact Diffuse	748.549	-1.965	4.150	1.450
	0.300	0.300	0.093	0.093
Ergodic	748.541	-1.958	4.150	1.450
	0.300	0.300	0.093	0.093

factor than using a fully diffuse prior (comparing second and third); since the AR(2) cycles have fairly small roots, that's probably not unexpected.

Table 3: Sinclair Estimates

	Approx Diffuse	Exact Diffuse	Conditional Ergodic	Full Ergodic
ϕ_1^y	0.743	0.743	0.744	0.747
ϕ_2^y	-0.266	-0.266	-0.289	-0.293
μ^y	0.842	0.842	0.845	0.846
ϕ_1^u	0.697	0.697	0.672	0.669
ϕ_2^u	-0.174	-0.174	-0.174	-0.175
F_{11}	1.453	1.453	1.427	1.410
F_{21}	-0.824	-0.824	-0.780	-0.763
F_{22}	0.496	0.496	0.478	0.475
F_{31}	-0.643	-0.643	-0.637	-0.629
F_{32}	0.068	0.068	0.033	0.028
F_{33}	0.238	0.238	0.254	0.253
F_{41}	0.607	0.607	0.585	0.575
F_{42}	-0.189	-0.189	-0.156	-0.151
F_{43}	-0.114	-0.114	-0.121	-0.121
F_{44}	-0.000	-0.000	0.000	-0.000

7.2 Example 2

The "baseline" system in Fabiani and Mestre (2004) has three observables: log of GDP, unemployment rate and inflation rate. The state equation takes the form:

$$\begin{bmatrix} y_t^* \\ \beta_t \\ (y-y^*)_t \\ (y-y^*)_{t-1} \\ u_t^* \\ \xi_t \\ (u-u^*)_t \\ (u-u^*)_{t-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \delta_1 & \delta_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_{t-1} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_t^{y*} \\ \varepsilon_t^{\beta} \\ \varepsilon_t^{yc} \\ \varepsilon_t^{u*} \\ \varepsilon_t^{\xi} \\ \varepsilon_t^{uc} \end{bmatrix}$$

where y_t^* is potential GDP, modeled as a local trend, u_t^* is the NAIRU, also modeled as a local trend, $(y-y^*)_t$ and $(u-u^*)_t$ are the output and unemployment gaps, respectively. The unemployment gap is modeled as an AR(2) process, while the output gap depends upon the lagged unemployment gap. Observed GDP is the sum (without error) of the latent y_t^* and $(y - y^*)_t$; similarly for observed unemployment. Observed inflation is modeled largely as a regression equation on lags of itself and lagged inflation in imports, with a loading from the lagged unemployment gap as the connection with the state space model. This has four unit roots (two sets of repeated roots) among eight states, with seven shocks: six in the state equation, plus a measurement error in the inflation equation.

The variances are quite small, so the approximate diffuse prior used in the paper was 10^4 variances, and the likelihood function was computed conditional on eight data points. Again, this is quite a bit more than is actually required, since with three observables, only three observations are needed to cover the eight states. As with the previous example, the smoothed means and variances are shown for observations 1, 4 and 10; because of the small scales of the variances, they are all multiplied by 10^6. For $t = 1$, the smoothed variance computed using the paper's approximation shows a rather catastrophic loss of precision. We also show this calculated with 10^2 initial variances, which is better behaved. Again, the mean isn't sensitive to the choice between these.

Table 7.2 shows a comparison of estimates, similar to what was done in the first example. The a_i, b_i and α are the regression coefficients from the inflation equation and ρ_2 is the loading for the unemployment gap on inflation. In this case, the loss of data points seems to be much more of an issue, probably because it's now five rather than one, and also because the AR(2) process for the unemployment gap has two fairly large roots, so the fully diffuse prior isn't as far off as it might be in other applications.

8 Conclusion

This paper has re-introduced and extended a result from the engineering literature for computing the initial variance for state-space models with a mixture of stationary and non-stationary roots. In addition to being capable of handling non-stationary states, it is also faster than the "textbook" method even for fully stationary models. We have also used empirical examples to document the possible problems with the use of simpler techniques for approximating diffuse priors.

Table 4: Fabiani-Mestre Model

	GDP Trend	GDP Gap	NAIRU
Time=1971:02			
10^4 Diffuse	13.498	0.007	0.024
	431.608	182661.632	−47950.302
10^2 Diffuse	13.498	0.007	0.024
	198.781	204.141	48.560
Exact Diffuse	13.498	0.007	0.024
	198.759	198.759	47.162
Ergodic	13.500	0.006	0.023
	127.017	127.017	30.804
Time=1972:01			
10^4 Diffuse	13.529	0.013	0.029
	159.160	159.160	36.863
10^2 Diffuse	13.529	0.013	0.029
	159.159	159.159	36.863
Exact Diffuse	13.529	0.013	0.029
	159.160	159.160	36.863
Ergodic	13.532	0.010	0.028
	107.117	107.117	25.205
Time=1973:02			
10^4 Diffuse	13.583	0.023	0.037
	101.749	101.749	23.796
10^2 Diffuse	13.583	0.023	0.037
	101.749	101.749	23.796
Exact Diffuse	13.583	0.023	0.037
	101.749	101.749	23.796
Ergodic	13.585	0.020	0.036
	75.568	75.568	18.110

Table 5: Fabiani-Mestre Estimates

	Approx Diffuse	Exact Diffuse	Conditional Ergodic	Full Ergodic
ϕ_1	-1.984	-1.986	-1.983	-1.968
ρ_2	-0.071	-0.071	-0.061	-0.065
δ_1	1.875	1.876	1.875	1.871
δ_2	-0.896	-0.897	-0.897	-0.892
a_1	-0.504	-0.505	-0.502	-0.541
a_2	-0.238	-0.238	-0.231	-0.244
a_3	-0.183	-0.183	-0.178	-0.195
b_1	0.077	0.077	0.077	0.077
b_2	0.061	0.061	0.061	0.065
b_3	0.036	0.036	0.036	0.039
b_4	0.075	0.075	0.075	0.074
α	-0.229	-0.229	-0.267	-0.151
$log\, \sigma^2_\pi$	-12.443	-12.443	-12.433	-12.318
$log\, \sigma^2_y$	-10.421	-10.421	-10.418	-10.441
$log\, \sigma^2_\beta$	-16.671	-16.673	-16.759	-16.689
$log\, \sigma^2_u$	-13.719	-13.718	-13.721	-13.778
$log\, \sigma^2_\xi$	-18.275	-18.275	-18.308	-18.356
$log\, \sigma^2_{uc}$	-14.723	-14.728	-14.755	-14.714

A Non-Standard Matrix Calculations

Non-standard matrix inversion was introduced into the statistics literature by Koopman (1997).[18] The goal is to compute an exact (limit) inverse of an input (symmetric) matrix of the form[19]

$$\mathbf{A} + \kappa\mathbf{B} \text{ as } \kappa \to \infty$$

The result will be the matrix of the form

$$\mathbf{C} + \mathbf{D}\kappa^{-1} + \mathbf{E}\kappa^{-2} \tag{14}$$

Note that the κ^{-2} term isn't needed in many applications. With the help of this expansion, it's possible to compute exact limits as $\kappa \to \infty$ for calculations like:

$$(\mathbf{F} + \mathbf{G}\kappa)\,(\mathbf{A} + \mathbf{B}\kappa)^{-1} \approx (\mathbf{F} + \mathbf{G}\kappa)\left(\mathbf{C} + \mathbf{D}\kappa^{-1}\right) \to \mathbf{FC} + \mathbf{GD}$$

You might think that you could just use a guess matrix of the form (14) multiply, match terms and be done quickly. Unfortunately, because matrices generally don't commute, it's very easy to get a left inverse or a right inverse, but not a true inverse. The matching terms idea is correct, but it has to be done carefully. We will start with a case that's more manageable. Note that this derivation is simpler than that in Koopman's paper, and will be easier to use for the proof in Appendix B. Here, the "infinite" matrix is isolated into an identity block, and we look only at the expansion through κ^{-1}.

$$\begin{aligned}
&\left(\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix} + \kappa \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\right) && \times \qquad (15) \\
&\left(\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}'_{12} & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & \mathbf{D}_{22} \end{bmatrix}\right) = && \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & \mathbf{I}_{n-r} \end{bmatrix} + \kappa^{-1} Rem
\end{aligned}$$

[18] *Non-standard analysis* was introduced into mathematics in the 1970's using results from "model theory" to embed the real numbers within a framework that includes "infinite" and "infinitesimal" numbers. It was hoped that this would allow simpler descriptions of limit calculations in real analysis, but never really caught on. See Blass (1978) for a brief survey of the ideas behind this.

[19] We're using conventional choices of **A**, **B** for the names of the matrices. These are not intended to match with the use of those names elsewhere in the paper.

Since the κ term in the product has to zero out, $\mathbf{C}_{11} = 0$ and $\mathbf{C}_{12} = 0$. Using that and collecting the $O(1)$ terms, we get

$$\left(\begin{bmatrix} \mathbf{D}_{11} & \mathbf{A}_{12}\mathbf{C}_{22} + \mathbf{D}_{12} \\ 0 & \mathbf{A}_{22}\mathbf{C}_{22} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & \mathbf{I}_{n-r} \end{bmatrix}$$

So we get $\mathbf{D}_{11} = \mathbf{I}_r, \mathbf{C}_{22} = \mathbf{A}_{22}^{-1}$ and $\mathbf{D}_{12} = -\mathbf{A}_{12}\mathbf{A}_{22}^{-1}$. $\mathbf{D}_{22}$ is arbitrary; it just ends up in the remainder, so for simplicity we can make it zero. If we check the reverse multiplication

$$\left(\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1}\begin{bmatrix} \mathbf{I}_r & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & 0 \end{bmatrix}\right)\left(\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix} + \kappa\begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\right)$$

we can verify that it also will be the identity + a term in κ^{-1}.

In general, we won't have an input matrix with the nice form in (15). However, for a p.s.d. symmetric matrix $\mathbf{B}$, we can always find a non-singular matrix $\mathbf{T}$ such that $\mathbf{TBT}'$ has that form.[20] So, in general, we can compute the inverse with:

$$(\mathbf{A} + \kappa\mathbf{B})^{-1} = \mathbf{T}'\left(\mathbf{TAT}' + \kappa\mathbf{TBT}'\right)^{-1}\mathbf{T} \tag{16}$$

For an input matrix pair $\{\mathbf{A}, \mathbf{B}\}$, this will produce an output pair $\{\mathbf{C}, \mathbf{D}\}$.

To derive the more accurate expansion in the case with the simpler form of $\mathbf{B}$, our test inverse is

$$\left(\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}'_{12} & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1}\begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & \mathbf{D}_{22} \end{bmatrix} + \kappa^{-2}\begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}'_{12} & \mathbf{E}_{22} \end{bmatrix}\right)$$

Everything from above goes through, except we now can't make $\mathbf{D}_{22}$ arbitrary. When we multiply out, the κ^{-1} terms are

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & \mathbf{D}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}'_{12} & \mathbf{E}_{22} \end{bmatrix} \text{or}$$

$$\begin{bmatrix} \mathbf{A}_{11}\mathbf{D}_{11} + \mathbf{A}_{12}\mathbf{D}'_{12} & \mathbf{A}_{11}\mathbf{D}_{12} + \mathbf{A}_{12}\mathbf{D}_{22} \\ \mathbf{A}'_{12}\mathbf{D}_{11} + \mathbf{A}_{22}\mathbf{D}'_{12} & \mathbf{A}'_{12}\mathbf{D}_{12} + \mathbf{A}_{22}\mathbf{D}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ 0 & 0 \end{bmatrix}$$

The bottom left element in the $\mathbf{AD}$ matrix is zero because of the first order solution. Since $\mathbf{D}_{22}$ was arbitrary from before, we can now solve for it as

$$\mathbf{D}_{22} = -\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\mathbf{D}_{12} = \mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}.$$

[20] The simplest to compute is based upon a modified version of the Choleski factorization.

With that, we also get

$$\mathbf{E}_{11} = -\mathbf{A}_{11}\mathbf{D}_{11} - \mathbf{A}_{12}\mathbf{D}'_{\mathbf{12}} = -\mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}$$

and

$$\mathbf{E}_{12} = -\mathbf{A}_{11}\mathbf{D}_{12} - \mathbf{A}_{12}\mathbf{D}_{22} = \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\right)\mathbf{A}_{12}\mathbf{A}_{22}^{-1}$$

$\mathbf{E}_{22}$ is now arbitrary. In terms of the input matrix, this is:

$$\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{A}_{22}^{-1} \end{bmatrix} + \kappa^{-1}\begin{bmatrix} \mathbf{I} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}} & \mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \end{bmatrix} +$$
$$\kappa^{-2}\begin{bmatrix} -\mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}} & \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\right)\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{\mathbf{12}}\right) & 0 \end{bmatrix} \tag{17}$$

The extension of this to the more general $\mathbf{B}$ matrix is as before. In the typical situation, the transforming $\mathbf{T}$ matrix will take the form

$$\begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix}$$

where $\mathbf{T}_1$ is $r \times n$ and $\mathbf{T}_2$ is $n - r \times n$. If that's the case, then the component matrices in the formula (17) are $\mathbf{A}_{11} = \mathbf{T}_1\mathbf{A}\mathbf{T}'_1$, $\mathbf{A}_{12} = \mathbf{T}_1\mathbf{A}\mathbf{T}'$ and $\mathbf{A}_{22} = \mathbf{T}_2\mathbf{A}\mathbf{T}'_2$. The expanded inverse will be (17) premultiplied by $\mathbf{T}'$ and postmultiplied by $\mathbf{T}$.

B Invariance to Diffuse Covariance Matrix

An important property of the diffuse prior is that the means and variances of the states, when conditioned on a set of observations sufficient to resolve the diffuseness, are independent of scaling and rotation of the diffuse covariance matrix. Note that this is *not* necessarily true until then: for the local trend model, if the diffuse covariance matrix is te identity, the first step in the Kalman filter will make both the level and the trend rate equal to $.5y_1$, while if it's $diag\{9.0, 1.0\}$, they will be $.9y_1$ and $.1y_1$ respectively. (The Kalman *smoothed* estimates will be the same with either, and and will be quite different from these). Because of the invariance, we can choose the simplest representation for that covariance matrix, which will typically be an identity submatrix in some transformation of the states. While this is fairly well-understood to be true by people working with diffuse priors, it has not, to our knowledge, been given a formal proof.

If the diffuse states are rank r, let one representation in the natural form of the model be $\Sigma_\infty = \mathbf{\Lambda}\mathbf{\Lambda}'$, where $\mathbf{\Lambda}$ is an $n \times r$ matrix. Any alternative which is diffuse in the same directions will take the form $\mathbf{\Lambda PP'\Lambda'}$ where $\mathbf{P}$ is a non-singular $r \times r$ matrix. Let a set of observations take the form $\mathbf{Y}_t = \mathbf{C'X}_t + \mathbf{v}_t$. Assume that $\mathbf{C'\Lambda}$ is full column rank. Although this will rarely be satisfied by the original representation of the state-space model, it's always possible to combine observation equations by successive substitution:

$$\mathbf{Y}_{t+1} = \mathbf{C'X}_{t+1} + \mathbf{v}_{t+1} = \mathbf{C'}\left(\mathbf{AX}_t + \mathbf{w}_t\right) + \mathbf{v}_{t+1} = \mathbf{C'AX}_t + \left(\mathbf{C'w}_t + \mathbf{v}_{t+1}\right)$$

similarly for higher values. While not useful in practice (since it defeats the recursive nature of the Kalman filter), it is useful for proving this result, since we need enough observables to resolve the diffuseness.

For fixed $\mathbf{\Lambda}$ and any choice for $\mathbf{P}$, the Kalman gain matrix takes the form:

$$\left(\mathbf{\Lambda PP'\Lambda'C}\kappa + \mathbf{G}\right)\left(\mathbf{C'\Lambda PP'\Lambda'C}\kappa + \mathbf{V}\right)^{-1} \tag{18}$$

where $\mathbf{G}$ and $\mathbf{V}$ are the finite parts of each factor which don't depend upon $\mathbf{P}$. Let

$$\begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} \mathbf{C'\Lambda\Lambda'C} \begin{bmatrix} \mathbf{T}_1' & \mathbf{T}_2' \end{bmatrix} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$$

for some matrices $\mathbf{T}_1$ $(r \times p)$ and $\mathbf{T}_2$ $(r - p \times p)$. This is a transformation matrix required in (16) when $\mathbf{P} = I_r$. Then $\mathbf{T}_1\mathbf{C'\Lambda}\left(\mathbf{T}_1\mathbf{C'\Lambda}\right)' = I_r$ and $\mathbf{T}_2\mathbf{C'\Lambda}\left(\mathbf{T}_1\mathbf{C'\Lambda}\right)' = 0$. Since $\left(\mathbf{T}_1\mathbf{C'\Lambda}\right)$ is non-singular, we must have $\mathbf{T}_2\mathbf{C'\Lambda} = 0$. Given a more general $\mathbf{P}$, the combination of $\mathbf{S}_1 = \left(\mathbf{T}_1\mathbf{C'\Lambda}\right)\mathbf{P}^{-1}\left(\mathbf{T}_1\mathbf{C'\Lambda}\right)^{-1}\mathbf{T}_1$ and $\mathbf{S}_2 = \mathbf{T}_2$ will work as the transforming matrix.

Using the transformation matrix $\mathbf{S}$, the block form of $\mathbf{SVS'}$ is

$$\begin{bmatrix} \mathbf{S}_1\mathbf{VS}_1' & \mathbf{S}_1\mathbf{VS}_2' \\ \mathbf{S}_2\mathbf{VS}_1' & \mathbf{S}_2\mathbf{VS}_2' \end{bmatrix}$$

The (limit) inverse in (16) takes the form:

$$\begin{bmatrix} \mathbf{S}_1' & \mathbf{S}_2' \end{bmatrix} \left(\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_2\mathbf{VS}_2'^{-1} \end{bmatrix} + \begin{bmatrix} I_r & \mathbf{D}_{12} \\ \mathbf{D}_{12}' & \mathbf{D}_{22} \end{bmatrix} \kappa^{-1} \right) \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix}$$

or

$$\mathbf{S}_2'\mathbf{S}_2\mathbf{VS}_2'^{-1}\mathbf{S}_2 + \left(\mathbf{S}_1'\mathbf{S}_1 + \mathbf{S}_2'\mathbf{D}_{12}'\mathbf{S}_1 + \mathbf{S}_1'\mathbf{D}_{12}\mathbf{S}_2 + \mathbf{S}_2'\mathbf{D}_{22}\mathbf{S}_2\right)\kappa^{-1}$$

The finite term in this is independent of $\mathbf{P}$ since $\mathbf{S}_2$ is the same for all $\mathbf{P}$. The interaction term between the κ term in the left factor of (18) and κ^{-1} in the inverse is

$$\left(\mathbf{\Lambda PP'\Lambda'C}\right)\left(\mathbf{S}_1'\mathbf{S}_1 + \mathbf{S}_2'\mathbf{D}_{12}'\mathbf{S}_1 + \mathbf{S}_1'\mathbf{D}_{12}\mathbf{S}_2 + \mathbf{S}_2'\mathbf{D}_{22}\mathbf{S}_2\right) \tag{19}$$

In the product, the last three terms drop out since $\mathbf{S}_2\mathbf{C}'\Lambda\mathbf{P} = 0$. (The third term is the transpose of the second). Thus, we're left with

$$\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\mathbf{S}_1'\mathbf{S}_1 = \boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\left(\mathbf{S}_1\mathbf{C}'\boldsymbol{\Lambda}\right)'\mathbf{S}_1 = \boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\left(\mathbf{T}_1\mathbf{C}'\boldsymbol{\Lambda}\mathbf{P}^{-1}\right)'\mathbf{S}_1 = \boldsymbol{\Lambda}\mathbf{P}\left(\mathbf{T}_1\mathbf{C}'\boldsymbol{\Lambda}\right)'\mathbf{S}_1 \tag{20}$$

where the third form uses $\mathbf{S}_1\mathbf{C}'\boldsymbol{\Lambda} = \mathbf{T}_1\mathbf{C}'\boldsymbol{\Lambda}\mathbf{P}^{-1}$. Plugging into this the definition of $\mathbf{S}_1$ and using the unitary nature of $\mathbf{T}_1\mathbf{C}'\boldsymbol{\Lambda}$, this collapses to $\boldsymbol{\Lambda}(\mathbf{T}_1\mathbf{C}'\boldsymbol{\Lambda})^{-1}\mathbf{T}_1$, independent of $\mathbf{P}$.

The formula for the conditional variance takes the form:

$$\left(\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\kappa + \mathbf{W}\right) - \left(\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\kappa + \mathbf{W}\right)\mathbf{C}\left(\mathbf{C}'\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\kappa + \mathbf{V}\right)^{-1}\mathbf{C}'\left(\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\kappa + \mathbf{W}\right)$$

The second order expansion of the inverse takes the form:

$$\left[\begin{array}{cc} \mathbf{S}_1' & \mathbf{S}_2' \end{array}\right]\left(\left[\begin{array}{cc} 0 & 0 \\ 0 & \mathbf{S}_2\mathbf{V}\mathbf{S}_2'^{-1} \end{array}\right] + \left[\begin{array}{cc} I_r & \mathbf{D}_{12} \\ \mathbf{D}_{12}' & \mathbf{D}_{22} \end{array}\right]\kappa^{-1} + \left[\begin{array}{cc} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{12}' & 0 \end{array}\right]\kappa^{-2}\right)\left[\begin{array}{c} \mathbf{S}_1 \\ \mathbf{S}_2 \end{array}\right]$$

The κ^0 terms in the correcting product will be:

$$\mathbf{W}\mathbf{C}\left[\begin{array}{cc} \mathbf{S}_1' & \mathbf{S}_2' \end{array}\right]\left(\left[\begin{array}{cc} 0 & 0 \\ 0 & \mathbf{S}_2\mathbf{V}\mathbf{S}_2'^{-1} \end{array}\right]\right)\left[\begin{array}{c} \mathbf{S}_1 \\ \mathbf{S}_2 \end{array}\right]\mathbf{C}'\mathbf{W}$$

$$\Lambda\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\kappa\left[\begin{array}{cc} \mathbf{S}_1' & \mathbf{S}_2' \end{array}\right]\left(\left[\begin{array}{cc} I_r & \mathbf{D}_{12} \\ \mathbf{D}_{12}' & \mathbf{D}_{22} \end{array}\right]\kappa^{-1}\right)\left[\begin{array}{c} \mathbf{S}_1 \\ \mathbf{S}_2 \end{array}\right]\mathbf{C}'\mathbf{W}$$

and its transpose, and

$$\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\kappa\left[\begin{array}{cc} \mathbf{S}_1' & \mathbf{S}_2' \end{array}\right]\left(\left[\begin{array}{cc} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{12}' & 0 \end{array}\right]\kappa^{-2}\right)\left[\begin{array}{c} \mathbf{S}_1 \\ \mathbf{S}_2 \end{array}\right]\mathbf{C}'\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'$$

The first of these is independent of choice of $\mathbf{P}$, since it depends only upon $\mathbf{S}_2$ and the second is just (19) post-multiplied by $\mathbf{C}'\mathbf{W}$. Since $\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\mathbf{S}_2' = 0$, the third term simplifies considerably to:

$$\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\left(\mathbf{S}_1'\mathbf{E}_{11}\mathbf{S}_1\right)\mathbf{C}'\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'$$

Now

$$\mathbf{E}_{11} = -\mathbf{S}_1\mathbf{V}\mathbf{S}_1' + \mathbf{S}_1\mathbf{V}\mathbf{S}_2'\left(\mathbf{S}_2\mathbf{V}\mathbf{S}_2'\right)^{-1}\mathbf{S}_2\mathbf{V}\mathbf{S}_1'$$

and we know from (20) that $\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}'\boldsymbol{\Lambda}'\mathbf{C}\mathbf{S}_1'\mathbf{S}_1$ is independent of $\mathbf{P}$. Since $\mathbf{S}_2$ is as well, this shows that the complete expression can be decomposed into the sum of products which aren't dependent upon $\mathbf{P}$.

References

BLASS, A. (1978): "Review of three books on non-standard analysis," *Bulletin of the American Mathematical Society*, 84(1), 34–41.

DURBIN, J., AND S. KOOPMAN (2002): *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.

FABIANI, S., AND R. MESTRE (2004): "A system approach for measuring the euro area NAIRU," *Empirical Economics*, 29(2), 311–341.

HAMILTON, J. (1994): *Time Series Analysis*. Princeton: Princeton University Press.

HARVEY, A. C. (1989): *Forecasting, structural time series models and the Kalman filter*. Cambridge: Cambridge University Press.

JOHANSEN, S. (2002): "A Small Sample Correction for the Test of Cointegrating Rank in the Vector Autoregressive Model," *Econometrica*, 70(5), 1929–1961.

KITAGAWA, G. (1977): "An Algorithm for Solving the Matrix Equation X = F X F' + S," *International Journal of Control*, 25(5), 745–753.

KOOPMAN, S. (1997): "Exact Initial Kalman Filtering and Smoothing for Nonstationary Time Series Models," *Journal of American Statistical Association*, 92(6), 1630–1638.

KOOPMAN, S., N. SHEPARD, AND J. A. DOORNIK (1999): "Statistical algorithms for models in state space using SsfPack 2.2," *Economics Journal*, 2, 113–166.

SINCLAIR, T. (2009): "The Relationships between Permanent and Transitory Movements in U.S. Output and the Unemployment Rate," *Journal of Money, Credit and Banking*, 41(2-3), 529–542.