A note on implementing the Durbin and Koopman simulation smoother

Jarocinski, Marek

European Central Bank

24 October 2014

A note on implementing the Durbin and Koopman simulation smoother

Marek Jarociński[*]

October 24, 2014

Abstract

The correct implementation of the Durbin and Koopman simulation smoother is explained. A possible misunderstanding is pointed out and clarified for both the basic state space model and for its extension that allows time-varying intercepts (mean adjustments).

Keywords: state space model; simulation smoother; trend output (*JEL:* C3; C15)

1 Introduction

Consider the state space model

$$y_t = Z_t \alpha_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, H_t) \tag{1a}$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t, \quad \eta_t \sim N(0, Q_t), \quad t = 1, ..., T, \quad \text{and} \tag{1b}$$

$$\alpha_1 \sim N(a_1, P_1), \tag{1c}$$

where y_t is the observation vector, α_t is the unobserved state vector, and ε_t and η_t are vectors of disturbances uncorrelated at all lags. The matrices $Z_t, H_t, T_t, R_t, Q_t, P_1$ and vector a_1 are assumed to be known. For further details and illustrations of this model see, e.g., Durbin and Koopman (2012).

This note explains the implementation of the Durbin and Koopman (2002) simulation smoother for this model, pointing out a possible misunderstanding. A simulation

[*] European Central Bank, Kaiserstrasse 29, 60311 Frankfurt am Main, Germany, e-mail: marek.jarocinski@ecb.int. I thank Gianni Amisano, Siem Jan Koopman and anonymous referees for very useful comments. The Matlab programs used for this note can be downloaded from runmycode.org.

smoother is an algorithm for drawing the states $\alpha = (\alpha_1', ..., \alpha_T')'$, or the disturbances $(\epsilon_1', \eta_1', ..., \epsilon_T', \eta_T)'$, from their distribution conditional on the observables $y = (y_1', ...y_T')'$. The misunderstanding may arise when drawing the states. It does not arise when drawing the disturbances.

2 The correct implementation

This section explains how to implement Durbin and Koopman's approach to drawing α conditional on y in the model (1a-1c). Let us call this algorithm 'Algorithm 2a' to differentiate it from their Algorithm 2.

Algorithm 2a. *(modified from Durbin and Koopman (2002) Algorithm 2, p.607)*

Step 1. Draw α^+ and y^+ by means of recursion (1a-1b), where the recursion is initialized by a draw $\alpha_1^+ \sim N(0, P_1)$.

Step 2. Construct the artificial series $y^* = y - y^+$ and compute $\hat{\alpha}^* = E(\alpha|y^*)$ by putting y^* through the Kalman filter and smoother.

Step 3. Take $\tilde{\alpha} = \hat{\alpha}^* + \alpha^+$. $\tilde{\alpha}$ is a draw from the distribution of α conditional on y.

An alternative implementation of this algorithm, which is also correct, uses (1a-1c) for the simulation of y^+, α^+ in Step 1 but then uses the model with $\alpha_1 \sim N(0, P_1)$ to compute the conditional expectation $\hat{\alpha}^* = E(\alpha|y^*)$ in Step 2.

The value added of this note lies in stating the above algorithm explicitly and in particular, in pointing out that a_1 needs to be reset to 0, i.e., the initial condition $\alpha_1 \sim N(a_1, P_1)$ (1c) needs to be replaced by $\alpha_1 \sim N(0, P_1)$ either in Step 1 or in Step 2. Durbin and Koopman (2002) state Algorithm 2, which is slower, and only suggest Algorithm 2a informally without stating it explicitly. In particular, they do not warn the reader that a_1 should be reset to 0 either in Step 1 or in Step 2, which gives rise to a possible misunderstanding that the unmodified model (1a-1c) can be used both in Step 1 and in Step 2.

Two conditions have a potential to render the above misunderstanding immaterial.

1. Diffuse initialization. Durbin and Koopman (2002) prove in their Appendix 2 that the diffuse elements of α_1^+ can be set equal to arbitrary quantities, hence the values of a_1 corresponding to these elements do not matter.

2. Zero mean. For the elements of α_1^+ that have a zero mean the correction obviously does not matter, since the corresponding values of a_1 equal 0 anyway.

Therefore, the misunderstanding is immaterial when all the elements of α_1 are either diffuse or have a zero mean.

In a model with intercepts another modification of Algorithm 2 is needed. Suppose the model is given by (1c),

$$y_t = d_t + Z_t\alpha_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, H_t) \quad \text{and} \tag{2a}$$

$$\alpha_{t+1} = c_t + T_t\alpha_t + R_t\eta_t, \quad \eta_t \sim N(0, Q_t), \tag{2b}$$

where d_t and c_t are intercepts that are known and may change over time. The remaining quantities are defined under equations (1a-1c). Algorithm 2a can also be used with this model, but the intercepts d_t and c_t should be reset to 0 for all t either in Step 1 or in Step 2.

3 A formal justification

I now provide a formal justification of Algorithm 2a. This algorithm assumes

$$\begin{pmatrix} \alpha \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_\alpha \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha y} \\ \Sigma_{\alpha y} & \Sigma_{yy} \end{pmatrix}\right) \quad \text{and} \quad \begin{pmatrix} \alpha^+ \\ y^+ \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha y} \\ \Sigma_{\alpha y} & \Sigma_{yy} \end{pmatrix}\right), \tag{3}$$

where the unconditional moments μ_α, μ_y, $\Sigma_{\alpha\alpha}$, $\Sigma_{\alpha y}$ and Σ_{yy} are functions of Z_t, H_t, T_t, R_t, Q_t, P_1, a_1 (c_t, d_t if applicable) implied by (1a-1c) or by (2a,2b,1c). Note, in particular, that resetting of μ_α and μ_y to 0 is achieved by resetting a_1 and, if applicable, c_t and d_t for all t to 0.

A draw $\tilde{\alpha}$ is generated as

$$\tilde{\alpha} = E(\alpha|y^*) + \alpha^+ = \mu_\alpha + \Sigma_{\alpha y}\Sigma_{yy}^{-1}(y - y^+ - \mu_y) + \alpha^+.$$

The first and second moments of $\tilde{\alpha}$ conditional on y are

$$E(\tilde{\alpha}|y) = \mu_\alpha + \Sigma_{\alpha y}\Sigma_{yy}^{-1}(y - \mu_y) = E(\alpha|y) \quad \text{and}$$
$$V(\tilde{\alpha}|y) = \Sigma_{\alpha y}\Sigma_{yy}^{-1}\Sigma_{yy}\Sigma_{yy}^{-1}\Sigma_{\alpha y}' - 2\Sigma_{\alpha y}\Sigma_{yy}^{-1}\Sigma_{\alpha y}' + \Sigma_{\alpha\alpha} = \Sigma_{\alpha\alpha} - \Sigma_{\alpha y}\Sigma_{yy}^{-1}\Sigma_{\alpha y}' = V(\alpha|y).$$

Hence, the first and second moments of $\tilde{\alpha}$ are correct and $\tilde{\alpha}$ is indeed a draw from $p(\alpha|y)$. Note, however, that setting the mean of (α^+, y^+) to (μ_α, μ_y) due to the discussed misunderstanding would have changed the value of $E(\tilde{\alpha}|y)$ and hence would have produced a draw from an incorrect density.

4 Numerical examples

I illustrate the effect of the possible misunderstanding using two numerical examples from the literature.

4.1 Nile data

The first example is the well-known local level model of the Nile data (a series of readings of the annual flow volume at Aswan from 1871 to 1970). Durbin and Koopman (2012), and others fit the following model to these data:

$$y_t = \alpha_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, 15099) \quad \text{and} \tag{4a}$$

$$\alpha_{t+1} = \alpha_t + \eta_t, \quad \eta_t \sim N(0, 1469.1), \tag{4b}$$

where y is the observed flow volume and α is its unobserved trend.

Table 1 reports the means and standard deviations of 10,000 draws of the trend α generated with several setups. "Initialization 1" assumes, for illustrative purposes, that α_1 comes from the Gaussian distribution centered at the first observation y_1 with the variance equal to the variance of the deviations of α from y throughout the sample, i.e. 15099. I generate 10,000 draws using Algorithm 2a and then I generate 10,000 draws with an incorrect variation of this algorithm, where I do not reset a_1 to 0 neither in Step 1 nor in Step 2. It is clear from Table 1 that the misunderstanding seriously distorts the simulation smoother: the mean of the trend in the first period, α_1, is 1114 with the correct algorithm (column 1) and 1350 with the incorrect variation (column 2). After 50 periods the initialization matters less and the means of the trend in period 50, α_{50}, obtained with Algorithm 2a and its incorrect variation are similar, 834 vs 835. Then I use the diffuse initialization for α and generate 10,000 draws first with Algorithm 2a and then with its incorrect variation. When the initialization is diffuse I obtain the same means and standard deviations of α with both implementations of Algorithm 2a, so I only report them once (column 3). To summarize, this numerical example illustrates that the misunderstanding can distort the results significantly 1) for the draws of states in the beginning of the sample and 2) when the initialization of the state is non-diffuse.

4.2 Trend of real GNP

The second example uses the model from Watson (1986). Watson fits the following model for the real Gross National Product (GNP) of the United States, y_t, observed quarterly

Table 1: Trend flow volume in the Nile model. Mean, standard deviation in parenthesis.

	Initialization 1 (non-diffuse)		Diffuse initialization
	Algorithm 2a	No resetting of a_1	
τ_1	1112 (57)	1350 (57)	1111 (64)
τ_{50}	834 (48)	835 (49)	835 (48)

from 1949 to 1984.

$$y_t = \tau_t + \varsigma_t, \tag{5a}$$

$$\tau_t = 0.008 + \tau_{t-1} + \eta_t^{\tau}, \quad \eta_t^{\tau} \sim N(0, 0.0057^2) \quad \text{and} \tag{5b}$$

$$\varsigma_t = 1.501\varsigma_{t-1} - 0.577\varsigma_{t-2} + \eta_t^{\varsigma}, \quad \eta_t^{\varsigma} \sim N(0, 0.0076^2), \tag{5c}$$

where τ_t is a trend and ς_t is a cycle, both unobservable.

Table 2 reports the mean and standard deviation of 10,000 draws of trend GNP, generated with several setups. “Initialization 1” assumes that ς_1 comes from the ergodic distribution of ς_t and that τ_1 is centered at the last value of GNP before the start of the sample, with the ergodic variance of ς_t. This is a natural assumption exploiting the stationarity of ς_t. I generate 10,000 draws using Algorithm 2a and then I generate 10,000 draws with an incorrect variation of this algorithm, where I do not reset a_1 and c_t to 0 neither in Step 1 nor in Step 2. It is clear from Table 2 that the misunderstanding seriously distorts the simulation smoother: the mean of the trend GNP in the first period, τ_1, is 6.24 with the correct algorithm (column 1) and 6.14 with the incorrect variation (column 2). After 50 quarters the initialization matters less and the means of the trend GNP in period 50, τ_{50}, obtained with Algorithm 2a and its incorrect variation are the similar, 6.66 vs 6.65. Next, I use the diffuse initialization of τ and ς. The mean of τ_1 is 6.28 with Algorithm 2a (column 3) and 6.09 with its incorrect variation (column 4). The misunderstanding matters here even with the diffuse initialization of τ and ς, because when model (5a-5c) is cast in form (1a-1b) the constant term of equation (5b) is a state with a non-zero and non-diffuse initialization and the failure to reset a_1 to 0 distorts the simulation smoother. Equivalently, when model (5a-5c) is cast in form (2a-2b) all the states are zero-mean or diffuse, but the failure to reset c_t to zero distorts the simulation smoother and yields the same numerical results. Again, after 50 quarters the trend GNP is similar with and without the misunderstanding.

Table 2: Trend GNP in Watson's model based on simulation smoothers. Mean, standard deviation in parenthesis.

	Initialization 1 (non-diffuse)		Diffuse initialization	
	Algorithm 2a	No resetting of a_1, c_t	Algorithm 2a	No resetting of a_1, c_t
τ_1	6.24 (0.02)	6.14 (0.02)	6.28 (0.03)	6.09 (0.03)
τ_{50}	6.66 (0.02)	6.65 (0.02)	6.66 (0.02)	6.65 (0.02)

5 Conclusion

This note explains the implementation of the Durbin and Koopman algorithm for drawing the states conditionally on the observables in a state space model, pointing out a possible misunderstanding. The misunderstanding matters when the initial state vector is not all zero-mean or diffuse, or when a nonzero intercept is present, and leads to incorrect draws of the states, especially in the beginning of a sample. By clarifying the possible misunderstanding, this note hopefully encourages an even wider use of the Durbin and Koopman algorithm by practitioners.

References

Durbin, J. and Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3):603–615.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. OUP Oxford.

Watson, M. W. (1986). Univariate detrending methods with stochastic trends. *Journal of Monetary Economics*, 18(1):49–75.