



CPB Netherlands Bureau for Economic
Policy Analysis

CPB Discussion Paper | 360

SMOOTHIES: A Toolbox for the Exact Nonlinear and Non-Gaussian Kalman Smoother

Joris de Wind

SMOOTHIES: A Toolbox for the Exact Nonlinear and Non-Gaussian Kalman Smoother*

Joris de Wind[†]

September 2017

Abstract

In this paper, I present a new toolbox that implements the exact nonlinear and non-Gaussian Kalman smoother for a wide class of discrete-time state space models, including models with implicit functions and equality constraints. The technical details are presented in an accompanying paper, while the toolbox is documented in the current one. The toolbox, which is built on top of Dynare, is very user-friendly and only requires the user to provide the state space model to be analyzed, while the toolbox automatically solves the smoothing problem. The toolbox can also be applied for conditional forecasting, which is demonstrated on the basis of a nonlinear macroeconomic model with forward-looking variables.

1 Introduction

The Kalman filter and smoother are essential tools for applied econometrics, with common applications in time series analysis such as trend-cycle decompositions and conditional forecasting. If a system is linear and Gaussian, the Kalman filter will deliver the best possible estimate of the current state of the system, while the Kalman smoother will update the past estimates in an optimal way. However, many state-of-the-art models in e.g. macroeconomics or finance are nonlinear and/or non-Gaussian, for which we need a generalization of the Kalman filter and smoother.

In this paper, I present a new toolbox that implements the exact nonlinear and non-Gaussian Kalman smoother that I have developed in an accompanying paper (De Wind, 2017). The toolbox can handle a wide class of discrete-time state space models, including models with implicit functions and equations without disturbances (e.g. equality constraints or identities). The toolbox can also deal with models with forward-looking variables, which

*I am very grateful to Michal Andrle, Junior Maih, and Rob van Harreveld for their comments and suggestions. Views expressed in this paper are my own and do not necessarily reflect those of the CPB.

[†]CPB Netherlands Bureau for Economic Policy Analysis, email: jorisdewind@gmail.com

are solved under the assumption of perfect foresight or through an user-provided function. This makes the toolbox well-suited for conditional forecasting exercises with nonlinear macroeconomic models, which may include implicit functions, identities as well as forward-looking variables.¹

The toolbox is built on top of Dynare, which itself is an open source Matlab toolbox that is widely used for macroeconomic models, but nevertheless well-suited for any discrete-time state space model.² The toolbox is very user-friendly and only requires the user to provide the state space model to be analyzed and needless to say the data. As a matter of fact, the user does not even have to distinguish between state and measurement equations, as this is automatically detected by the toolbox based on the provided data. The toolbox automatically sets up the optimization problem that forms the basis of the exact nonlinear and non-Gaussian Kalman smoother, constructs the first-order conditions using analytical derivatives, and solves the system with the stacked-time algorithm.³

The next section provides a description of the exact nonlinear and non-Gaussian Kalman smoother, which basically follows De Wind (2017), while additional implementation details are given in a subsequent section. The paper also contains a separate section with two examples, one of which is a conditional forecasting exercise that is related to the ‘observation procedure’ of Sandee, Don, and van den Berg (1984).

2 Exact nonlinear and non-Gaussian Kalman smoother

This section provides a brief description of the exact nonlinear and non-Gaussian Kalman smoother and is largely based on De Wind (2017). Although the toolbox developed in the current paper does not require users to put their model in state space form, the exposition here is based on such representation.

Consider the nonlinear and non-Gaussian state space model

$$\begin{aligned} Z(y_t, \alpha_t, \varepsilon_t) &= 0 & \varepsilon_t &\sim p_\varepsilon(\varepsilon_t) \\ T(\alpha_t, \alpha_{t-1}, \eta_t) &= 0 & \eta_t &\sim p_\eta(\eta_t) & t = 1, \dots, n \\ & & \alpha_0 &\text{ is given} \end{aligned} \quad (1)$$

where y_t is a vector with observables and α_t is a vector with state variables. $Z(\cdot)$ and $T(\cdot)$ could in principle be time-varying functions and also the distributions of the disturbances are

¹The toolbox can be downloaded from www.github.com/jorisdewind.

²The toolbox requires that Dynare is installed as well. Dynare can be downloaded from www.dynare.org.

³I would like to acknowledge that the toolbox makes use of many components of Dynare, including the model parser, analytical derivatives, and the stacked-time algorithm. Furthermore, the stacked-time algorithm is explained in e.g. Juillard (1996) and Hollinger (1996), while a brief description is given in the appendix of De Wind (2017).

allowed to be time-varying, yet time subscripts are omitted here for notational convenience. As a matter of fact, if one wishes to account for uncertainty about the initial state of the system, this should be incorporated in the above representation by appropriately adjusting the distribution of the state disturbance in the first period.⁴ The above representation is very general and allows for implicit functions as well as equations without disturbances (e.g. equality constraints or identities).

The state vector α_t captures the underlying dynamics of the state space system, but it is not directly observed. For linear state space models with Gaussian disturbances, the Kalman filter and smoother are the preferred means to estimate the state vector α_t based on the vector with observables y_t . The Kalman filter and smoother differ from each other in the amount of conditioning information that is used for the estimation. That is, the Kalman filter estimates the state vector α_t based on the available information up to time t exclusive (y^{t-1}), whereas the Kalman smoother estimates the state vector α_t based on all the available information in the sample period (y^n).

Nevertheless, the current paper is concerned with state space models that are nonlinear and/or non-Gaussian, for which we cannot employ the standard Kalman filter and smoother recursions. In an accompanying paper I have developed an exact treatment of the Kalman smoother for models that nonlinear and/or non-Gaussian, see De Wind (2017). One slight difference with the standard Kalman smoother is that my approach delivers the mode of the state of the system as opposed to the mean.⁵ In particular, the idea is to maximize the likelihood of the disturbances subject to the constraints implied by the model, which can be formalized by the following Lagrangian⁶

$$\mathcal{L}(\varepsilon^n, \eta^n, \alpha^n, \Lambda_Z^n, \Lambda_T^n) = \sum_{t=1}^n \left(\log(p_\varepsilon(\varepsilon_t)) + \log(p_\eta(\eta_t)) + \Lambda_{Z,t}' Z(y_t, \alpha_t, \varepsilon_t) + \Lambda_{T,t}' T(\alpha_t, \alpha_{t-1}, \eta_t) \right) \quad (2)$$

where $\Lambda_{Z,t}$ and $\Lambda_{T,t}$ are vectors with Lagrangian multipliers for the measurement and state equations, respectively. The first-order conditions with respect to $\Theta^n = \{\varepsilon^n, \eta^n, \alpha^n, \Lambda_Z^n, \Lambda_T^n\}$

⁴Time-varying functions and distributions can be modeled most easily using a time dummy variable.

⁵In the linear case with Gaussian disturbances the mean and mode are actually the same.

⁶This maximization problem yields the mode of the state of the system. The key point is that maximizing the complete data log-likelihood function with respect to the state vector is equivalent to maximizing the conditional density of the state vector. This is explained in De Wind (2017).

are given by

$$\begin{aligned}
& \hat{\alpha}_0 = \alpha_0 \\
& \frac{\partial \log(p_\varepsilon(\varepsilon_t))}{\partial \varepsilon_t} \Big|_{\Theta^n = \hat{\Theta}^n} + \hat{\Lambda}'_{Z,t} \frac{\partial Z(y_t, \alpha_t, \varepsilon_t)}{\partial \varepsilon_t} \Big|_{\Theta^n = \hat{\Theta}^n} = 0 \\
& \frac{\partial \log(p_\eta(\eta_t))}{\partial \eta_t} \Big|_{\Theta^n = \hat{\Theta}^n} + \hat{\Lambda}'_{T,t} \frac{\partial T(\alpha_t, \alpha_{t-1}, \eta_t)}{\partial \eta_t} \Big|_{\Theta^n = \hat{\Theta}^n} = 0 \\
& \hat{\Lambda}'_{Z,t} \frac{\partial Z(y_t, \alpha_t, \varepsilon_t)}{\partial \alpha_t} \Big|_{\Theta^n = \hat{\Theta}^n} + \hat{\Lambda}'_{T,t} \frac{\partial T(\alpha_t, \alpha_{t-1}, \eta_t)}{\partial \alpha_t} \Big|_{\Theta^n = \hat{\Theta}^n} \\
& \quad + \hat{\Lambda}'_{T,t+1} \frac{\partial T(\alpha_{t+1}, \alpha_t, \eta_{t+1})}{\partial \alpha_t} \Big|_{\Theta^n = \hat{\Theta}^n} = 0 \quad t = 1, \dots, n \\
& \hat{\Lambda}_{T,n+1} = 0 \\
& Z(y_t, \hat{\alpha}_t, \hat{\varepsilon}_t) = 0 \\
& T(\hat{\alpha}_t, \hat{\alpha}_{t-1}, \hat{\eta}_t) = 0
\end{aligned} \tag{3}$$

The above system of equations can be solved by e.g. the stacked-time algorithm.⁷ This yields the mode of the state vector $\hat{\alpha}^n$, i.e. the main output of the exact nonlinear and non-Gaussian Kalman smoother.

Based on this approach, I have developed a toolbox which only requires the user to provide the state space model to be analyzed and needless to say the data. As a matter of fact, the user does not even have to distinguish between state and measurement equations, as this is automatically detected by the toolbox based on the provided data. The toolbox automatically sets up the appropriate optimization problem, constructs the first-order conditions using analytical derivatives, and solves the resulting system of equations with the stacked-time algorithm.⁸

3 Implementation details

Users need to provide a standard Dynare model file with the model equations and of course a file with the data for the observables.⁹ Users do not have to distinguish between state and measurement equations, as this is automatically detected by the toolbox based on the provided data. Users are allowed to provide a dataset with missing observations, so that

⁷The stacked-time algorithm is explained in e.g. Juillard (1996) and Hollinger (1996), while a brief description is given in the appendix of De Wind (2017).

⁸The toolbox is freely available at www.github.com/jorisdewind but requires a Matlab licence (though the toolbox is likely to work under Octave as well). The toolbox is developed on top of Dynare, which itself is an open source Matlab toolbox, see www.dynare.org. I would like to acknowledge that the toolbox makes use of many components of Dynare, including the model parser, analytical derivatives, and the stacked-time algorithm.

⁹See Adjemian et al. (2011) for the Dynare reference manual or the user guide referenced therein.

the toolbox can be used for conditional forecasting exercises, problems with ragged-edge datasets, and the treatment of mixed-frequency data.

The toolbox is built on top of Dynare and makes use of many components of Dynare, including the model parser, analytical derivatives, and the stacked-time algorithm. The toolbox automatically sets up the optimization problem that forms the basis of the exact nonlinear and non-Gaussian Kalman smoother, constructs the first-order conditions using analytical derivatives, and solves the system with the stacked-time algorithm. Each step requires a run of Dynare.

The toolbox first executes an initial run of the Dynare parser to prepare various objects, from which the toolbox extracts information to construct the Lagrangian function. After that, the toolbox executes a second run of the Dynare parser which yields the Jacobian matrix with partial derivatives. The needed partial derivatives are extracted from the Jacobian matrix to construct the first-order conditions, which are finally solved by a third run of Dynare. The details of this three-stage procedure are documented within the source code, which is available through GitHub.¹⁰

4 Examples

This section contains two examples. The first example is just for demonstration purposes, while the second example is a conditional forecasting exercise with a nonlinear macroeconomic model. The latter example is interesting on its own for practitioners at e.g. central banks and international institutions. The same examples are also given in De Wind (2017), although the technical details are left out in the current paper.

4.1 Nonlinear state space model

Consider the following nonlinear Gaussian state space model

$$\begin{aligned}
 y_t &= \cos(x_t^2 + \varepsilon_t) \exp(z_t) & \varepsilon_t &\sim N(0, h_y^2) \\
 x_t &= (1 - \rho_x) \mu_x + \rho_x x_{t-1} + \eta_t^x & \eta_t^x &\sim N(0, q_x^2) \\
 z_t &= (1 - \rho_z) \mu_z + \rho_z z_{t-1} + \eta_t^z & \eta_t^z &\sim N(0, q_z^2) & t = 1, \dots, n \quad (4) \\
 & x_0 \text{ and } z_0 \text{ are given}
 \end{aligned}$$

The goal is to estimate the state variables given the data (and for given parameter values). The exact nonlinear Kalman smoother solves this estimation problem in an optimal manner and my toolbox can be employed for this purpose.

¹⁰www.github.com/jorisdewind

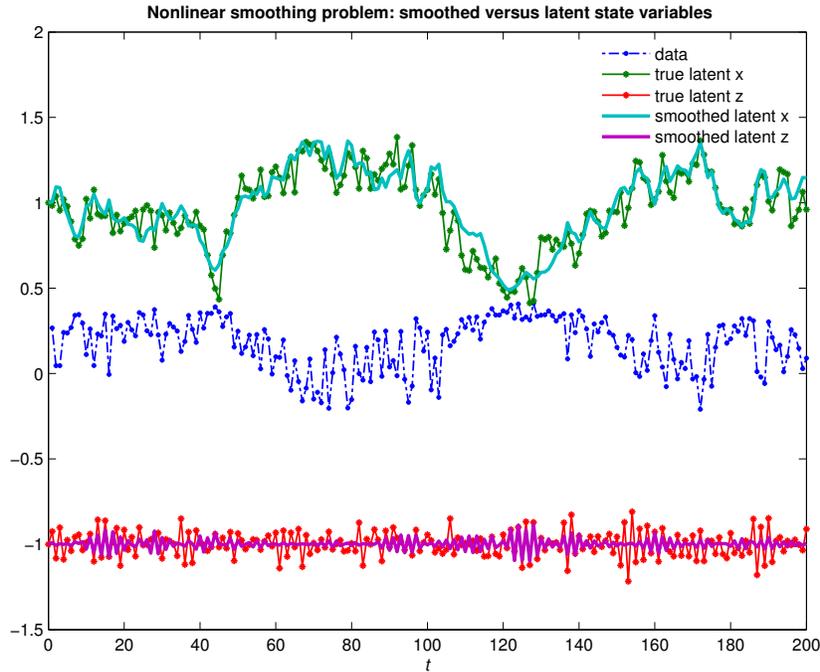


Figure 1: Numerical example with measurement error

Numerical example with measurement error. I have generated artificial data with the following parameter configuration: $\rho_x = 0.95$, $\mu_x = 1$, $\rho_z = -0.75$, $\mu_z = -1$, $h_y = 0.25$, $q_x = 0.1$, and $q_z = 0.05$. The number of periods is set to 200 and the state vector is initialized at its unconditional mean, that is $x_0 = \mu_x$ and $z_0 = \mu_z$.

The observable data and latent state variables are plotted in figure 1 together with the smoothed state variables. It is clear from the figure that the state variable x_t is very well identified and estimated very well over the whole sample period, while the state variable z_t is considerably more difficult to track. Of course, that is due to the model (with three disturbances and only one measurement per time period) since the exact nonlinear Kalman smoother delivers the best possible estimate of the state variables given the data.

Numerical example without measurement error. I have also applied the exact nonlinear Kalman smoother to the same artificial data but then absent of measurement error, i.e. $h_y = 0$.

The observable data and latent state variables are plotted in figure 2 together with the smoothed state variables. The latent state variables are exactly the same as in the previous example, while the observable data are a little bit smoother. Now the state variable x_t is captured almost perfectly, while the state variable z_t is sometimes still difficult to track.

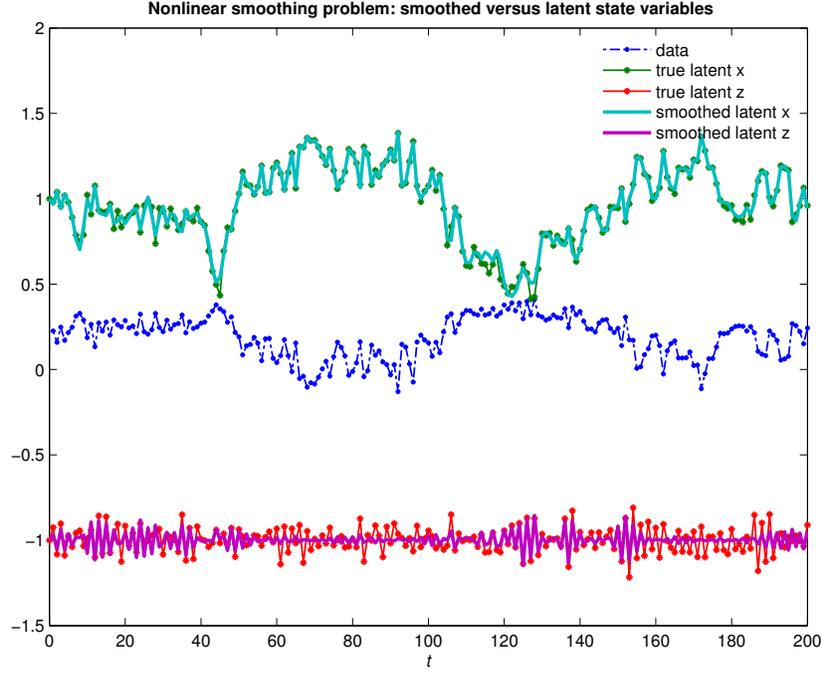


Figure 2: Numerical example without measurement error

Again, that is due to the model since the exact nonlinear Kalman smoother delivers the best possible estimate of the state variables given the data.

4.2 Conditional forecasting

Consider the following dynamic version of the IS-LM model

$$\begin{aligned}
 Y_t &= C_t + I_t + G_t \\
 Y_t^d &= Y_t - T_t \\
 T_t &= t_0 + t_1 Y_t + \varepsilon_t^T \\
 C_t &= c_0 + c_1 Y_{t-1}^d + c_2 Y_t^d + c_3 \mathbb{E}_t [Y_{t+1}^d] + c_4 R_t + c_5 R_t^2 + \varepsilon_t^C \\
 I_t &= i_0 + i_1 Y_{t-1} + i_2 Y_t + i_3 \mathbb{E}_t [Y_{t+1}] + i_4 R_t + i_5 R_t^2 + \varepsilon_t^I \\
 M_t^d &= m_0 + m_1 Y_t + m_2 R_t + m_3 R_t^2 + \varepsilon_t^{MD} \\
 M_t^d &= M_t^s
 \end{aligned} \tag{5}$$

The first equation is the aggregate resource constraint which equates domestic output Y_t to the sum of consumption C_t , investment I_t , and government spending G_t . The second equa-

tion defines disposable income Y_t^d as the difference between income Y_t and taxes T_t . The third equation determines the level of taxes as a function of income, implying that taxes are determined endogenously as opposed to government spending which is an exogenous variable. The fourth equation determines consumption as a function of lagged, current, and expected future disposable income as well as the interest rate R_t . The fifth equation determines investment as a function of lagged, current, and expected future output and also the interest rate. The sixth equation determines money demand M_t^d which depends on output and the interest rate. The seventh equation determines the money market equilibrium, where money supply M_t^s is determined exogenously. Finally, the ε_t^X variables are the residuals of the behavioral equations, which are assumed to be independently normally distributed.

It is straightforward to run a deterministic simulation (assuming perfect foresight) for the purpose of making a forecast for given trajectories of the exogenous variables, e.g. using the stacked-time algorithm. However, we are often also interested in making a forecast conditional on trajectories for one or more of the endogenous variables in order to incorporate the views of an expert or satellite model (or we might have a ragged-edge dataset). The exact nonlinear and non-Gaussian Kalman smoother can be employed for making such *conditional* forecasts.¹¹

Suppose we want to make a forecast conditional on the assumption that the government budget closes for the next eight periods, that is $T_t = G_t$ for $t = 1, \dots, 8$. Of course, the tax equation in the above IS-LM model will yield non-zero residuals but since taxes also appear in the other model equations it is very likely that the other residuals will be non-zero too. The idea behind the exact nonlinear Kalman smoother is to maximize the likelihood of the residuals subject to the constraints implied by the model, which delivers the most likely conditional forecast. This optimization problem is solved automatically by my toolbox.¹²

As an illustration, figure 3 plots the unconditional forecast for output together with the conditional forecasts conditioning on $T_t = G_t$ for two, four, and eight conditioning periods, respectively. For the chosen parameterization, taxes are higher in the conditioning path than what the model endogenously implies, so that the conditional forecasts for output are lower the longer the conditioning period.

¹¹My conditional forecasting approach is related to the ‘observation procedure’ of Sandee, Don, and van den Berg (1984), and can be considered as a generalization of their approach. See De Wind (2017) for a more detailed comparison between the two approaches.

¹²The forward-looking variables are solved under the assumption of perfect foresight, requiring the terminal condition that the economy returns to its steady state in the long run. In principle it is also possible for users to treat the forward-looking variables differently through an user-provided function.

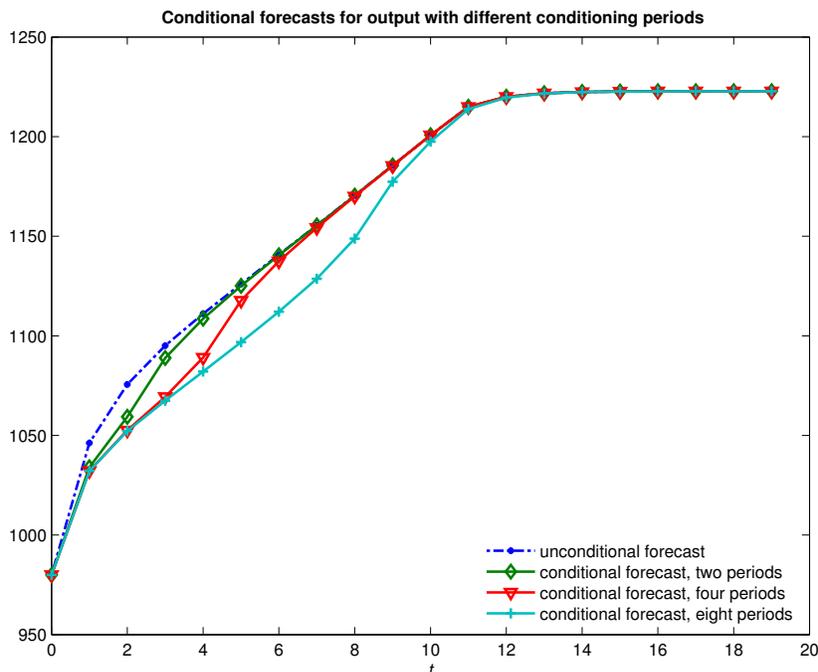


Figure 3: Example conditional forecasting with a nonlinear macroeconomic model

References

- ADJEMIAN, S., H. BASTANI, M. JUILLARD, F. KARAMÉ, F. MIHOUBI, G. PERENDIA, J. PFEIFER, M. RATTO, AND S. VILLEMOT (2011): “Dynare: Reference Manual, Version 4,” Dynare Working Papers, 1, CEPREMAP.
- DE WIND, J. (2017): “Exact Nonlinear and Non-Gaussian Kalman Smoother for State Space Models with Implicit Functions and Equality Constraints,” CPB Discussion Paper No. 359.
- HOLLINGER, P. (1996): “The Stacked-Time Simulator in TROLL: A Robust Algorithm for Solving Forward-Looking Models,” unpublished manuscript.
- JUILLARD, M. (1996): “DYNARE: a program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm,” CEPREMAP Working Papers, 9602, CEPREMAP.
- SANDEE, J., F. J. H. DON, AND P. J. C. M. VAN DEN BERG (1984): “Adjustment of projections to recent observations,” *European Economic Review*, 26, pp. 153–166.



Publisher:

CPB Netherlands Bureau for Economic Policy Analysis
P.O. Box 80510 | 2508 GM The Hague
T (088) 984 60 00

September 2017