ISSN 1440-771X
ISBN 0 7326 1069 9

M O N A S H   U N I V E R S I T Y

AUSTRALIA

Understanding the Kalman Filter: An Object Oriented Programming Perspective

Ralph D. Snyder and Catherine S. Forbes

Working Paper 14/99

December 1999

DEPARTMENT OF ECONOMETRICS AND BUSINESS STATISTICS

Understanding the Kalman Filter: An Object Oriented Programming Perspective

Ralph D. Snyder

Department of Econometrics and Business Statistics

Monash University

Clayton, Victoria 3185

Australia

(613) 9905 2366

ralph.snyder@buseco.monash.edu.au

Catherine S. Forbes

Department of Econometrics and Business Statistics

Monash University

Clayton, Victoria 3185

Australia

(613) 9905 2471

catherine.forbes@buseco.monash.edu.au

Keywords

time series analysis, forecasting, Kalman filter, dynamic linear statistical models, object oriented programming.

Abstract

The basic ideas underlying the Kalman filter are outlined in this paper without direct recourse to the complex formulae normally associated with this method. The novel feature of the paper is its reliance on a new algebraic system based on the first two moments of the multivariate normal distribution. The resulting framework lends itself to an object-oriented implementation on computing machines and so many of the ideas are presented in these terms. The paper provides yet another perspective of Kalman filtering, one that many should find relatively easy to understand.

1. Introduction

An understanding of the Kalman filter (Kalman and Bucy, 1961) is important for those with an interest in time series analysis. It provides an essential building block for the process used to estimate unknown parameters associated with any linear time series model based on normally distributed disturbances. For trial values of the parameters, it is applied recursively to successive values of a time series, to yield a sequence of one-step ahead predictions. These outputs of the filter are best, linear, unbiased predictors, a consequence being that the associated one-step ahead errors are orthogonal. The Kalman filter conveniently produces these one-step ahead prediction errors, together with their variances. It may be viewed as a device to unravel the dependencies in any time series. As it is always possible to construct this sequence of errors from the original series and vice-versa, both series contain the same information from an estimation perspective. Thus the likelihood associated with the trial parameter values can be evaluated directly from the orthogonal one-step ahead prediction errors. The advantage of this strategy is that explicit inversion of the large variance matrix associated with the original series is avoided.

Despite its central role, the Kalman filter is not widely understood by the statistical profession. It is rarely taught in undergraduate statistics courses. And those who study it at a graduate level often emerge with, at best, a moderate understanding of the method. This problem was recognised a few decades ago by Duncan and Horn (1972). Their strategy was to couch the derivation of the filter in terms of the more familiar regression framework. Meinhold and Singpurwalla (1983) also attempted a resolution. They employed the theory of conditional probability in what they term a Bayesian approach. Both attempts provide interesting insights into the method. Yet the problem still seems to persist to this day.

The cause of this pedagogic problem is open to debate. Referencing our own experience as students of the technique, we attribute it to the large and unwieldy sets of notation, together with the relatively complex recursive formulae found in most expositions. The contribution of this paper is to simplify matters by introducing and relying on a novel algebraic system centred on the first two moments of a multivariate normal distribution. The concepts underpinning this algebraic system are presented mainly in terms of the language of object oriented programming.

The plan of this paper is as follows. The algebraic system is introduced in Section 2. State space models are reviewed, building on the algebraic system, in Section 3. The Kalman filter, in its new form, is presented in Section 4.

2. Moment object and Associated Methods

The Kalman filter is essentially a mechanism for calculating the means and the variances of multivariate normal probability distributions at different points of time. It is therefore natural to focus on the first two moments of the normal distribution and to treat them as a mathematical object with associated operations and methods. The Kalman filter may then be described in terms of the resulting algebraic system.

This algebraic system is outlined in a pseudo-code similar to the Matlab programming language. Matlab is fast evolving into the default standard for mathematically oriented programming languages. One difference, however, is that we continue to use subscripts to avoid an excessive use of brackets, a luxury that is currently not available to computer programmers. Most of the notation in the pseudo-code is self-explanatory. We highlight a few important conventions:

- the elementary data type is a matrix and all conventional matrix operations apply

- if *a, b, c, d* are matrices then [*a b*; *c d*] designates the matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
- zeros(*m*,*n*) is a function that returns an $m \times n$ matrix of zeros
- inv(*a*) returns the inverse of the matrix *a*
- *i:j* represents the index sequence $i,i+1,\ldots,j$ where $j>i$
- $x(i_1{:}i_2, j_1{:}j_2)$ is the specified submatrix of *x*
- comments in the code follow the % symbol.

We adopt the practice of using the same character to represent a random vector and a data structure containing its first two moments. To avoid potential ambiguity, the random vector is distinguished by a tilde. For example $\underset{\sim}{x}$ represents a random vector with a data structure *x*.

The fields of the data structure, denoted by $x.m$ and $x.v$, contain the mean and variance respectively.

A data structure is created and tagged as a moment object by the constructor function shown in Insert 1. It should be emphasised that in practice additional code may be required to check data and to undertake other housekeeping tasks. These details are excluded to ensure that the focus of the paper is dedicated to core concepts only.

```
function x = moments(m, v)
x.m = m                          % assign mean to its field
x.v = v                          % assign variance to its field
x = class(x, 'moments')          % tag x as a moments object
```

Insert 1. Constructor function for the moments object

Some common operations are overloaded to work on moment objects.

Addition

The generic function for addition in the pseudo-code is assumed to be *plus*. If two commensurate normal random vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ are uncorrelated, the function as it applies to moment objects, is shown in Insert 2. Once in place, quasi-mathematical statements such as

$$z = x + y$$

invoke the plus function when *x* and *y* are moment objects.

```
function z =plus(x,y)
z = moments(x.m + y.m, x.v + y.v)
```

Insert 2. Function for adding moments objects

Multiplication

The generic function for multiplication is assumed to be *times*. If *a* is a matrix and *x* is a commensurate moment object then the function required to overload the multiplication operator is shown in Insert 3. This function is called when a quasi-mathematical statement of the form

$y = a * x$

is encountered in the computer code.

```
function y = times(a, x)
y = moments(a * x.m, a * x.v * a' )
```

Insert 3. Function for premultiplying moments objects by a matrix

Conditioning Operator

The interdependencies between the elements of a normally distributed random vector are completely summarised by the covariances in the associated variance matrix. If some elements of the random vector are observed, then this should indirectly provide information about those related elements that are not observed. Suppose the elements are arranged so that all those that are observable are placed at the top of the random vector. Then the typical random vector $\underset{\sim}{x}$ contains sub-vectors $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ commensurate with this arrangement. It is to be expected that the additional information in the observed value $\bar{x}_1$ of $\underset{\sim}{x}_1$ should reduce the variance of $\underset{\sim}{x}_2$.

The distribution of $\underset{\sim}{x}_2$ conditional on $\underset{\sim}{x}_1 = \bar{x}_1$ summarises the impact of the new information. The moments of this conditional distribution are represented by $x_2|\bar{x}_1$. They are obtained using a standard result from the theory of the multivariate normal distribution - see, for example, Anderson (1958).

The means and variances of $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ are designated by $m_1,\ m_2,\ v_{11},\ v_{22}$ respectively. Furthermore, their covariances are represented by the matrix v_{21}. The required result is that the revised distribution has a mean $m_2 + v_{21}v_{11}^{-1}\left(\underset{-}{x}_1 - m_1\right)$ and variance $v_{22} - v_{21}v_{11}^{-1}v_{12}$. Note the reduction in the variance induced by the term $v_{21}v_{11}^{-1}v_{12}$, which is positive if $v_{12} \neq 0$. Furthermore, the revised distribution of $\underset{\sim}{x}_1$ is concentrated entirely on the single point $\bar{x}_1$.

One can think of | as a conditioning operator. Languages like Matlab, however, often reserve this symbol for the logical operator *or*. There is no reason to restrict its use in this way. In the context of moment objects it can be defined as the conditioning operator without eliminating its use as a logical operator in a more general context. Assuming that it is invoked by the symbol |, the generic function 'or'outlined Insert 4, performs the required conditioning. For reasons that will become clearer later, it is convenient to consider the distribution of the entire revised random vector $\underset{\sim}{x}|\bar{x}_1$.

It is now possible to write quasi-mathematical statements in computer code like

$x = x \mid \bar{x}_1$.

The effect is to revise the moment object of a random vector $\underset{\sim}{x}$ to encompass the information on the sub-vector $\underset{\sim}{x}_1$ provided by the observation $\bar{x}_1$.

```
function $y$ = or($x$, $\bar{x}_1$)
$m = x.m$
$v = x.v$
$n$ = length($m$)
$r$ = length($\bar{x}_1$)
$m_1 = m(1{:}r)$
$m_2 = m(r+1{:}n)$
$v_{11} = v(1{:}r, 1{:}r)$
$v_{22} = v(r+1{:}n, r+1{:}n)$
$m$ = [ $\bar{x}_1$ ; $m_2 + v_{21}$ * inv($v_{11}$) * ( $\bar{x}_1 - m_1$)]
$v$ = [zeros($r$, $n$); zeros($n-r$, $r$) , $v_{22} - v_{21}'$ * inv($v_{11}$) * $v_{21}$]
$y$ = moments($m$, $v$)
```
Insert 4. ‘Or’ function to perform conditioning operation

The moment objects and the above operations form an algebraic system. The addition of uncorrelated moment objects possesses the closure, commutative and associative properties. Multiplication between pairs of moment objects is not allowed. Multiplication, as presented in this paper, is defined in the narrower sense of an operation between a matrix and a moment object. This operation is not commutative. Also, moment objects have no unique inverses. Thus the algebraic system is not an algebra.

3. Data Generating Process

In describing the data generating process we shall make use of the concept of a variant array, terminology that is borrowed from Microsoft’s Visual Basic programming language. A variant is a generic data type. It may contain a number, a string, a date or any user defined data type or object. A rectangular array of variants is similar in many respects to a spreadsheet. In Matlab they are referred to as cell arrays.

Multiple time series, associated with typical period *t*, will be designated by a random *k*-vector $\underset{\sim}{x}(t)$. The collection of all such vectors, designated by $\underset{\sim}{x}$, is an $n \times 1$ variant array. In our context, $\underset{\sim}{x}(t)$ strictly represents the contents of the *t*th cell of the array, the contents being a random vector. The elements of $\underset{\sim}{x}(t)$ are the components of the process.

It is assumed that the evolution of the process is governed by the first-order recurrence relationship

$$\underset{\sim}{x}(t) = A\underset{\sim}{x}(t-1) + B\underset{\sim}{u}(t), \qquad (2.1)$$

where $\underset{\sim}{u}(t) \sim \mathrm{NID}(0,\Omega)$, *A* and *B* are fixed matrices and Ω is a fixed positive semi-definite variance matrix. It is also assumed that the disturbances $\underset{\sim}{u}(t)$ are uncorrelated with the so-called seed vector $\underset{\sim}{x}(0)$.

As above, the random vector of components is partitioned into sub-vectors:

- $\underset{\sim}{x_1}(t)$ - the observable components which form the time series of interest
- $\underset{\sim}{x_2}(t)$ - the unobserved components such as growth rates and seasonal effects.

Example 1

A univariate time series, represented by a random variable $\underset{\sim}{y}(t)$ in typical period *t*, is governed by a so-called local level data generating process (Muth, 1960) when the process mean $\underset{\sim}{l}(t)$ follows a random walk. More specifically

$$\underset{\sim}{y}(t) = \underset{\sim}{l}(t) + \underset{\sim}{u}_1(t) \qquad \text{(measurement equation)}$$

$$\underset{\sim}{l}(t) = \underset{\sim}{l}(t-1) + \underset{\sim}{u}_2(t) \qquad \text{(transition equation)}$$

where $\underset{\sim}{u_1}(t)$ and $\underset{\sim}{u_2}(t)$ are mutually and individually contemporaneously and inter-temporarily uncorrelated. This is the simplest member of the structural time series collection of models advocated by Harvey (1991) in place of the more traditional Box-Jenkins approach (Box, Jenkins and Reinsel,1976). Using the transition equation to eliminate $\underset{\sim}{l}(t)$ from the measurement equation we get, in conformity with (2.1), the matrix equation

$$\begin{bmatrix} \underset{\sim}{y}(t) \\ \underset{\sim}{l}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{y}(t-1) \\ \underset{\sim}{l}(t-1) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{u_1}(t) \\ \underset{\sim}{u_2}(t) \end{bmatrix}.$$

Note that $\underset{\sim}{l}(t)$ is the unobserved component.

Example 2

The single disturbance version of the local level model, a special case of the framework in Snyder (1985), and Ord, Koehler and Snyder (1997), is

$$\underset{\sim}{y}(t) = \underset{\sim}{l}(t) + \underset{\sim}{u}(t) \qquad \text{(measurement equation)}$$

$$\underset{\sim}{l}(t) = \underset{\sim}{l}(t-1) + \alpha\,\underset{\sim}{u}(t) \qquad \text{(transition equation)}$$

where $\underset{\sim}{u}(t)$ is a scalar. This model underlies simple exponential smoothing (Brown, 1959). It can be written as

$$\begin{bmatrix} \underset{\sim}{y}(t) \\ \underset{\sim}{l}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{y}(t-1) \\ \underset{\sim}{l}(t-1) \end{bmatrix} + \begin{bmatrix} 1 \\ \alpha \end{bmatrix} \underset{\sim}{u}(t)$$

in conformity with the structure of (2.1).

A statistical description entails finding the distributions of $\underset{\sim}{x}(t)$ over time. When none of the elements of the $\underset{\sim}{x}(t)$ are observable, the formulae for recursive calculation of the moments of $\underset{\sim}{x}(t)$, designated by $\mu(t)$ and $\Sigma(t)$ for the mean and variance respectively, are usually stated as

$$\mu(t) = A\mu(t-1) \tag{2.2}$$

$$\Sigma(t) = A\Sigma(t-1)A' + B\Omega B'. \tag{2.3}$$

Such formulae may be used when the moments of the seed random vector $\underset{\sim}{x}(0)$ are specified.

Using moment objects, these equations may be replaced by

$$x(t) = A * x(t-1) + B * u\,. \qquad (2.4)$$

This parsimonious statement is sufficient to invoke the operations necessary to compute the required moments. A striking feature of (2.4) is that it has the same structure as the original data generating process (2.1). The random variables have simply been replaced by their corresponding moment objects. The effect is to consolidate the specification of the algorithm required to undertake the necessary calculations. Another feature of the equation is that the time subscript has been dropped from the term representing the moments of the disturbances. In our version of the dynamic linear data generating process, it has been assumed that the moments of the disturbances remain unchanged over time. The time subscript for the moment object *u* is redundant, something that is not true of the original disturbances $\underset{\sim}{u}(t)$.

4. Kalman Filter and Associated Methods

The Kalman filter applies to those situations where some of the elements of the random vector $\underset{\sim}{x}(t)$ are observable. In the preceding local level examples, $\underset{\sim}{y}(t)$ is observable but $\underset{\sim}{l}(t)$ is not. In the following, it is convenient to use $y(t)$ as an alias for $\underset{\sim}{x_1}(t)$. Adaptations of the formula (2.4) are required to revise the moments of $\underset{\sim}{x}(t)$ to reflect the informational impact of successive observations $\bar{y}(1),\ldots,\bar{y}(s)$ where *s*, a time period, is potentially different from *t*. We denote the moments of $\underset{\sim}{x}(t)$ conditional on $\bar{y}(1),\ldots,\bar{y}(s)$ by the variant array element $x(t,s)$. The two-dimensional variant array of resulting conditional moments is depicted in Table 1 for the case $n = 4$. The moments in column 0 are obtained without recourse to any data. They correspond to the moments in the previous section. The moments on the diagonal are often referred to as filtered quantities. Those in the final column 4 correspond to the so-called smoothed quantities.

	Periods of data				
Period (t)	0	1	2	3	4
0	$x(0,0)$	$x(0,1)$	$x(0,2)$	$x(0,3)$	$x(0,4)$
1	$x(1,0)$	$x(1,1)$	$x(1,2)$	$x(2,2)$	$x(1,4)$
2	$x(2,0)$	$x(2,1)$	$x(2,2)$	$x(2,2)$	$x(2,4)$
3	$x(3,0)$	$x(3,1)$	$x(3,2)$	$x(3,2)$	$x(4,4)$
4	$x(4,0)$	$x(4,1)$	$x(4,2)$	$x(4,2)$	$x(4,4)$

Table 1. Notation for moments of distributions incorporating data

The Kalman filter is geared to finding the filtered quantities on the diagonal. In the process it also makes use of the moments in the band immediately below the diagonal. The elements in this band correspond to the moments of the one-step ahead prediction distributions. The function for the Kalman filter, in terms of the moment objects, is shown in Insert 5. Here $\bar{y}$

denotes a one-dimensional variant array, the *t*th cell containing the vector of time series observations associated with period *t*, and *x*0 is the moment object for the distribution of the components in period 0. The output of this function corresponds to the moment object $x(n,n)$. Note that the consolidation of distributional information into moment objects has provided a relatively simple specification of the filter.

```
function $x$ = Filter( $\bar{y}$ , $A$, $B$, $u$, $x0$)
$n$ = length($y$)              %sample size
$x$ = $x0$                     %moments of seed distribution
for $t$ = 1:$n$
   $x = A * x + B * u$         %one-step ahead prediction
   $x = x|\bar{y}(t)$          %revision with sample information
end
```

Insert 5. Kalman filter function

An alternative to the Kalman filter is to use a regression approach. Equations (2.1) are stacked, ensuring that all unobserved components on the left-hand side are transferred to the right-hand side of the associated equations. This gives a regression problem, as originally proposed by Duncan and Horn (1972), where all the unobserved components make up the vector of unknown regression coefficients. The estimates of the coefficients, obtained using the least squares methods, depend on the entire sample of size *n*. Thus they correspond to the smoothed values depicted in the final column of Table 1. Duncan and Horn (1972) never recommended the use of regression to estimate dynamic linear models. Instead, they used the regression as a stepping-stone in the derivation of the Kalman filter. It appears that they may have been dissuaded from the regression calculations by the large size of the resulting problem. The matrices associated with the regression are sparse, however, so the approach is now feasible using special sparse matrix solvers.

As illustrated in Table 1 for the special case *n* = 4, the Kalman filter and regression approaches overlap in the sense that they both yield the moments of the distribution of $\underset{\sim}{x}(n,n)$. This is the information required to generate the moments of the prediction distributions using the relationships, expressed in terms of moment objects, as:

$$x(n+j,n) = A * x(n+j-1,n) + B * u \qquad j = 1,2,\ldots$$

As mentioned earlier, the Kalman filter is often used to generate the orthogonal one-step ahead prediction errors and their variances for evaluating the likelihood function (Schweppe, 1965). The Kalman filter, as described in Insert 5, can be supplemented with statements to calculate and store the required quantities, as shown in Insert 6.

```
function [e, v] = Filter($\bar{y}$, A, B, u, x0)
n = length(y)                 %sample size
r = length($\bar{y}(1)$)      %number of observable time series
x = x0                        %moments of seed distribution
for t = 1:n
   $x = A * x + B * u$        %moments of one-step ahead prediction distribution
   $\hat{y} = x.m(1{:}r)$     %one-step ahead prediction
   $v(t) = x.v(1{:}r,1{:}r)$  %variance matrix of one-step ahead error
   $e(t) = \bar{y}(t) - \hat{y}$  %one-step ahead prediction error
   $x = x|\bar{y}(t)$         %revision with sample information
end
```

Insert 6. Kalman filter function: the augmented version

There are several approaches regarding the initialisation of the filter. If the time series under consideration are stationary, then the Kalman filter is typically seeded with the marginal distribution of the components vector. The mean of this distribution is zero. The variance matrix *V* satisfies the linear equation

$$V = AVA' + B\Omega B' .$$

A method for solving this equation is described in Harvey (1991).

For nonstationary time series, the marginal distribution is undefined. A common practice, in this particular case, is to use a diagonal variance matrix with large but finite numbers as a substitute for what should be infinite variances. An approximation of this kind may not always work well. Then specialised algorithms are required (Ansley and Kohn, 1985; de Jong, 1991; Snyder and Saligari, 1996; Koopman and Durbin , 1999). The details of this particular problem go beyond the scope of the present paper. Alternatively, the moments of the seed distribution can be chosen using Bayesian subjective prior probability considerations.

6. Conclusions

It has been shown that moment objects are a convenient way of consolidating information pertaining to multivariate normal distributions. By defining appropriate addition, multiplication and conditioning operators, the Kalman filter was presented in a more concise form than usual. The consequent simplification of the specification of the Kalman filter should have significant pedagogic advantages, making it more amenable to a broader base of students of statistics.

The exposition was focused on the invariant form of the state space model where *A*, *B* and Ω do not change over time. The invariance property was only used to reduce notational overheads. Thus the theory is easily adapted to more general cases where these matrices depend on time.

The algebraic system developed in this paper transcends the theory of Kalman filtering. We expect it to have useful applications to other areas of statistics that are built on the multivariate normal distribution.

References

Anderson, T.W. 1958. *An Introduction to Multivariate Statistical Analysis*, New York: John Wiley.

Ansley, C.F. and R. Kohn, “Estimation, filtering and smoothing in state space models with incompletely specified initial conditions”, *Ann. Stat.* 13, 1286-1316.

Box, G.E.P, Jenkins,G.M. and Reinsel. G. C. (1976), *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day.

Brown, R.G. 1959. *Statistical Forecasting for Inventory Control*, New York: Mc.Graw-Hill.

De Jong, P. (1991), “The diffuse Kalman filter”, *Ann. Stats.*, 19, 1073-1083.

Duncan, D. B., and S. D. Horn (1972), “Linear dynamic recursive estimating from the viewpoint of regression analysis”, *Journal of the American Statistical Association*, 67, 815-821.

Harvey, A. C. (1991), *Forecasting, Structural Time Series and the Kalman Filter*, Cambridge: Cambridge University Press.

Kalman, R.E. (1960) “A new approach to linear filtering and prediction problems”, *Journal of Basic Engineering, Transactions ASME, Series D*, 82, 35-45.

Kalman, R.E. and R.S. Bucy. 1961. ‘New results in linear filtering and prediction theory’, *Journal of Basic Engineering, Transactions ASME, Series D*, 83, 95-108.

Kaminsky, P.G., Bryson, A.E. and Schmidt, S.F. (1971) “Discrete square root filtering: a survey of current techniques”, *IEEE Trans. Autom. Control AC-16*, 727-736.

Koopman, S.J. and J.Durbin (1999) “Filtering and smoothing of state vector for diffuse state space models”, http://www.econ.cam.ac.uk/faculty/harvey/papers.htm

Meinhold, R. J. and Singpurwalla, N. D. (1983) “Understanding the Kalman Filter”, *The American Statistician*, 37, 123-127.

Muth, J.F. (1960) “Optimal properties of exponentially weighted forecasts”, *Journal of the American Statistical Association*, 55, 299-305.

Ord, J.K. , Koehler, A. B. and Snyder, R. D. (1997) “Estimation and prediction for a class of dynamic nonlinear statistical models”, *Journal of the American Statistical Association*, 92, 1621-1629.

Schweppe, F. (1965), “Evaluation of likelihood functions for Gaussian signals,” *IEEE Transactions on Information Theory’*, 11, 61-70.

Snyder, R. D. (1985), ‘Recursive estimation of dynamic linear statistical models’, *Journal of the Royal Statistical Society B*, 47, 272-276.

Snyder, R. D., and Saligari, G. (1996), “Initialisation of the Kalman filter with partially diffuse initial conditions*”, Journal of Time Series Analysis*, 17, 409-424.