Manual StateSpace: Time series analysis of variable seasonal signals

Taco Broerse
Delft University of Technology

October 24, 2014

version 2.0

The purpose of the MATLAB program *StateSpace* is to estimate seasonal signals and long term variations from time series. Instead of the classical deterministic approach where seasonal signals with fixed amplitudes and phases and linear trends are estimated, the state space method is implemented that allows for variations in time of the seasonal signals and trends.

1 Theory

1.1 State space formulation

We follow the state space method of the book of Durbin & Koopman: Time series analysis by state space methods in the description of the time series (Durbin and Koopman, 2012). We describe our univariate time series y_t (a [n x 1] vector) as:

$$y_t = Z\alpha_t + \epsilon_t \tag{1}$$

$$\epsilon_t \sim N(0, H) \tag{2}$$

where Z is the design matrix that links observations to the [m x 1] state α, the vector with m unobserved variables that is allowed to develop with time for epochs $t = 1, 2.., n$. The irregular term ϵ is assumed to be normally distributed with variance H and zero mean. The standard version of the steady state model is commonly defined for equal time steps, which leads to integer time when time is normalised. However, the program can also be used for continuous time, meaning that any time step can be used. Time t is normalised by

$$t = \frac{time - time_1}{T_{obs}} + 1; \tag{3}$$

with T_{obs}, the (average) periodicity of the sampling

$$T_{obs} = \frac{time_n - time_1}{n - 1} \tag{4}$$

Missing observations for integer time t can be handled as well.

We compute the state α in a recursive manner

$$\alpha_{t+1} = T\alpha_t + \eta_t \qquad \eta_t \sim N(0, Q) \tag{5}$$

where we make the state stochastic by adding random walk term η. Process noise variance Q is assumed to be independent from H. T is the transition matrix that describes how the state changes from epoch to epoch.

We can explicitly model several components in the time series model, in the program the following are included:

$$y_t = \mu_t + c_t + \epsilon_t \tag{6}$$

where μ_t is the level component and c_t is the (seasonal harmonic) cycle component. Note that seasonal has commonly a different meaning in steady state modeling, so we will refer to the cycle component rather than seasonal component.) We can write these components recursively, which is for the level:

$$\mu_{t+1} = \mu_t + \nu_t + \zeta_t \qquad \zeta_t \sim N(0, \sigma_\zeta^2) \tag{7}$$

with slope ν_t

$$\nu_{t+1} = \nu_t + \xi_t \qquad \xi_t \sim N(0, \sigma_\xi^2) \tag{8}$$

To create a smooth trend we set $\sigma_\zeta^2 = 0$, which leads to the so-called local trend model or integrated random walk model. This model can be extended by quadratic terms as well.

The cycle component with period T_{cycle}

$$c_t = c \cdot cos(\lambda t) + c^* sin(\lambda t) \tag{9}$$

with

$$\lambda = \frac{2\pi}{T_{cycle}} T_{obs} \tag{10}$$

can be written recursively (using a little magic of the addition theorems for $cos(t+dt)$ and $sin(t+dt)$) such that the recursion becomes independent of time when dt is integer (equation 3.2.4 of Durbin and Koopman (2012)):

$$c_{t+1} = c_t \cdot cos(\lambda) + c_t^* \cdot sin(\lambda) + \omega_t \qquad \omega_t \sim N(0, \sigma_\omega^2) \tag{11}$$

$$c_{t+1}^* = -c_t \cdot sin(\lambda) + c_t^* \cdot cos(\lambda) + \omega_t^* \qquad \omega_t^* \sim N(0, \sigma_{\omega^*}^2) \tag{12}$$

Phases ϕ and amplitudes A can be extracted from c_t and c_t^*

$$A_t = \sqrt{c_t^2 + c_t^{*2}} \tag{13}$$

$$\phi_t = -tan^{-1} \frac{c_t^*}{c_t} - t\lambda \quad mod(2\pi) \tag{14}$$

For nonhomogeneously spaced data (continuous time) the same expressions can be used, in that case the arguments of the harmonics become $(\lambda \cdot dt)$. Furthermore all variances will be scaled by dt (Harvey, 1990).

The state vector α [m x 1] thus becomes

$$\alpha_t = [\mu_t \; \nu_t \; \omega_{1,t} \; \omega^*_{1,t} \; .. \; \omega_{j,t} \; \omega^*_{j,t}]' \tag{15}$$

where we apply j cycle terms. The transition matrix then becomes:

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 & .. & 0 & 0 \\ 0 & 1 & 0 & 0 & .. & 0 & 0 \\ 0 & 0 & cos(\lambda_1) & sin(\lambda_1) & .. & 0 & 0 \\ 0 & 0 & -sin(\lambda_1) & cos(\lambda_1) & .. & 0 & 0 \\ .. & .. & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & 0 & .. & cos(\lambda_j) & sin(\lambda_j) \\ 0 & 0 & 0 & 0 & .. & -sin(\lambda_j) & cos(\lambda_j) \end{bmatrix} \tag{16}$$

for continuous time:

$$T = \begin{bmatrix} 1 & dt & 0 & 0 & .. & 0 & 0 \\ 0 & 1 & 0 & 0 & .. & 0 & 0 \\ 0 & 0 & cos(\lambda_1 dt) & sin(\lambda_1 dt) & .. & 0 & 0 \\ 0 & 0 & -sin(\lambda_1 dt) & cos(\lambda_1 dt) & .. & 0 & 0 \\ .. & .. & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & 0 & .. & cos(\lambda_j dt) & sin(\lambda_j dt) \\ 0 & 0 & 0 & 0 & .. & -sin(\lambda_j dt) & cos(\lambda_j dt) \end{bmatrix} \tag{17}$$

The design matrix:

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & .. & 0 & 0 \\ 0 & 0 & 0 & 0 & .. & 0 & 0 \\ 0 & 0 & 1 & 0 & .. & 0 & 0 \\ 0 & 0 & 0 & 0 & .. & 0 & 0 \\ .. & .. & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & 0 & .. & 1 & 0 \\ 0 & 0 & 0 & 0 & .. & 0 & 0 \end{bmatrix} \tag{18}$$

Process noise variance Q [m x m] and irregular component variance H [n x n] are:

$$Q = I\sigma^2_\eta \tag{19}$$

$$H = I\sigma^2_\epsilon \tag{20}$$

with

$$\sigma_\eta = [\sigma_\zeta \; \sigma_\xi \; \sigma_{\omega_1} \; \sigma_{\omega^*_1} \; .. \; \sigma_{\omega_j} \; \sigma_{\omega^*_j}]' \tag{21}$$

for continuous time we get:

$$Q = I\sigma^2_\eta dt \tag{22}$$

$$H = I\sigma^2_\epsilon dt \tag{23}$$

Using the state space method we can estimate the state vector α, its error variance matrix V, plus disturbance ϵ_t and process noise η_t using a Kalman filter and smoother.

1.2 Kalman filter

The Kalman filter estimates from $Y_t = y_1, .., y_t$

$$a_{t+1} = E(\alpha_{t+1}|Y_t) \tag{24}$$

and

$$P_{t+1} = var(\alpha_{t+1}|Y_t) \tag{25}$$

for $t = 1, ..n$ where the Kalman filter consists of the five equations Durbin and Koopman (2012):

$$\begin{aligned}
&v_t = y_t - Z_t a_t \qquad F_t = Z_t P_t Z_t' + H_t \\
&K_t = T_t P_t Z_t'(F_t)^{-1} \\
&a_{t+1} = T_t a_t + K_t v_t \quad P_{t+1} = T_t P_t (T_t - K_t Z_t)' + R_t Q_t R_t'
\end{aligned} \tag{26}$$

where v_t is the prediction error with variance F_t and K_t is the Kalman gain; R_t is commonly an identity matrix.

1.3 State smoother

The smoother consists of a backward loop for $t = n, ..1$ (chapter 4.4.4 of Durbin and Koopman (2012)), which provides the smoothed state $\hat{\alpha_t}$ and its error variance V_t

$$\begin{aligned}
&r_{t-1} = Z_t' F_t^{-1} v_t + L_t' r_t \quad N_{t-1} = Z_t' F_t^{-1} Z_t + L_t' N_t L_t \\
&L_t = T_t - K_t Z_t \\
&\hat{\alpha_t} = a_t + P_t r_{t-1} \qquad V_t = P_t - P_t N_{t-1} P_t
\end{aligned} \tag{27}$$

where r_t is called the weighted sum of innovations and N_t the weighted sum of the inverse variances and $r_n = 0$ and $N_n = 0$.

1.4 Disturbance smoother

An alternative smoother computes smoothed estimates of disturbances ϵ and η (chapter 4.5 of Durbin and Koopman (2012)):

$$\hat{\epsilon_t} = E(\epsilon_t|Y_n) \quad \hat{\eta_t} = E(\eta_t|Y_n) \tag{28}$$

These are calculated using the following recursion for $t = n, ..1$ by:

$$\begin{aligned}
&u_t = F_t^{-1} v_t - K_t' r_t && D_t = F_t^{-1} + K_t' N_t K_t \\
&N_{t-1} = Z_t' D_t Z_t + T_t' N_t T_t - Z_t' K_t' N_t T_t - T_t' N_t K_t Z_t && \\
&\hat{\epsilon_t} = H_t u_t && Var(\epsilon_t|Y_n) = H_t - H_t D_t H_t \\
&\hat{\eta_t} = Q_t R_t' r_t && Var(\eta_t|Y_n) = Q_t - Q_t R_t' N_t' R_t Q_t
\end{aligned} \tag{29}$$

1.5 Estimation of additional parameters

Until now, we have assumed disturbance variances to be known (process noise variances in Q and variance of the irregular component in H). These variances are in normal circumstances unknown and can be estimated using the same Kalman filter and smoother by maximising the likelihood $L(Y_n)$.

$$L(Y_n) = p(y_1, ..., y_n) \tag{30}$$

More in specific, $logL$ is maximised, which equals (chapter 7.2 of Durbin and Koopman (2012)):

$$logL(Y_n) = -\frac{n}{2}log(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\left(log|F_t| + v_t'F_tv_t\right) \tag{31}$$

We maximise $logL$ by changing disturbance parameters ϵ and η, using the EM algorithm (that stands for expectation-maximisation) and, while not the fastest optimisation algorithm around, should always lead to increasing log likelihood. Koopman (1993) provides equations for updates of time constant σ_η^2 and σ_ϵ^2 for η with dimensions larger than one:

$$\bar{\sigma}_\epsilon^2 = \tilde{\sigma}_\epsilon^2 + \frac{1}{n}\tilde{\sigma}_\epsilon^2\sum_{t=1}^{n}\left(u_t^2 - D_t\right)\tilde{\sigma}_\epsilon^2 \tag{32}$$

$$\bar{\sigma}_\eta^2 = \tilde{\sigma}_\eta^2 + \frac{1}{n-1}\tilde{\sigma}_\eta^2\sum_{t=1}^{n}\left(r_{t-1}^2 - N_{t-1}\right)\tilde{\sigma}_\eta^2 \tag{33}$$

where u_t, D_t, r_t and N_t are determined using the disturbance filter. The EM-algorithm is applied until convergence of $logL$ has been reached. The EM-algorithm searches for a local optimum only, which means that it is important to start with an educated guess for the disturbance parameters.

2 Program use

The program is provided together with an example script **run_state_space.m** that reads data, sets settings, executes the state space analysis and makes figures. This example script can be adapted according to the needs of the user, or can serve as an example how to integrate the program in existing scripts. **run_state_space.m** calls the function **statespaceanalysis.m** that performs the actual state space analysis. This function can be used as is, but needs three inputs to work properly:

- *tseries* a structure containing tseries.time (time vector) and *tseries.y* (observations vector), both of size [n x 1]
- *settings* a structure containing settings, describing the model that should be applied to the data as well as some settings for the program itself, see **SetDefaults.m** for defaults.
- *distvar* a structure containing initial settings for the disturbance variance parameters (ϵ and η), see **SetDefaults.m** for defaults.

The program **statespaceanalysis.m** calls a number of subroutines, in the following order:

FindMissingEpochs.m In case of equal time steps, checks the occurrence of gaps in the time series.

StateSpacePreProcess.m Preprocessing of data: normalisation of the time vector

EM.m Estimation/Maximisation algorithm that in a statistical sense optimally determines disturbance variances. This subroutine can be by-passed when variances are manually chosen.

InitialiseState.m Initialisation of the state and its error variance for the first epoch (no observations included). By default $a_1 = 0$, $P_1 = 10^7$. *

StateSpaceSetup.m Definition of state space model. Produces the matrices that will be used in the Kalman filter and smoother.

Kalman.m Runs Kalman filter (forward pass)

Smoother.m Runs smoother (backward pass)

DisturbanceSmoother.m Runs disturbance smoother (backward pass).

PostProcessState.m Restores components from the state vector (that was determined by Smoother.m)

ExtractVariance.m Extracts variance for components from state error variance matrices (that was determined by Smoother.m)

LogLikeliHood.m Computes the log likelihood

* n.b. The program does not include diffuse initialisation yet.

Furthermore, there are a few useful optional subroutines:

LeastSquaresTrend.m Determining least squares fit, using the same functional model as the Kalman filter.

SpectralAnalysis.m Spectral analysis using the MATLAB FFT function for inspection of frequency components. For equally spaced data only.

lomb.m Lomb-Scargle periodigram for inhomogeneously spaced data sets. Note: periodigrams are not normalized, in contrast to the original lomb.m code.

PostProcessTrend.m Computes a mean slope from the time variable slope. Warning: the error variance of the mean slope does not include correlations between errors of subsequent slope estimates and hence provides overly optimistic values. There is still an issue with the covariance matrix of the state vector, which has to be solved in a future release of *StateSpace*.

synthetictimeseries produces a synthetic time series.

2.1 Inputs and settings

2.1.1 Inputs regarding observations

Required input includes:

- *tseries.time*: the time vector of size [n x 1].
- *tseries.y*: the observation vector of size [n x 1], examples are provided in **run_state_space.m**.

Further settings, related to the observations, that need the user's attention are:

- *settings.continuoustime* if false, equally spaced time steps are required. However, since continuous time is the general case, any dataset can be run using *settings.continuoustime* = true.

- *settings.checktime* may be used such that **StateSpacePreProcess.m** checks whether *tseries.time* is indeed equally spaced, however the current implementation may be to strict.

2.1.2 State space functional model settings

- *settings.slope*: option for inclusion of a slope.
- *settings.acc*: option for inclusion of a quadratic term.
- *settings.cycle*: option for inclusion of one or more cycle terms. If *settings.cycle* is set to true, *settings.periodscycle* is the [1 x j] matrix with the respective periods of the cycle terms.
- *settings.intervention*: option for including a step discontinuity. If true supply *settings.interventiontime* for time of the intervention.
- *settings.regression*: option for including regression, i.e. calculating the correlation of the time series with the time series of the regressor. If true, supply *tseries.regressors* of size [n x 1] or [m x n] in case of m regressors.

2.1.3 Initial values for disturbance variance

- *distvar.irr*: σ_ϵ^2 (irregular component); size [1 x 1]
- *distvar.level*: σ_ζ^2 (level component); size [1 x 1]
- *distvar.slope*: σ_ξ^2 (trend component); size [1 x 1]
- *distvar.acc*: $\sigma_{\xi_2}^2$ (quadratic component); size [1 x 1]
- *distvar.cycle*: σ_ω^2 and $\sigma_{\omega^*}^2$ (cycle components); size [1 x j] or [2 x j]. In case a [1 x j] array is supplied, σ_ω^2 and $\sigma_{\omega^*}^2$ are assumed to be equal.

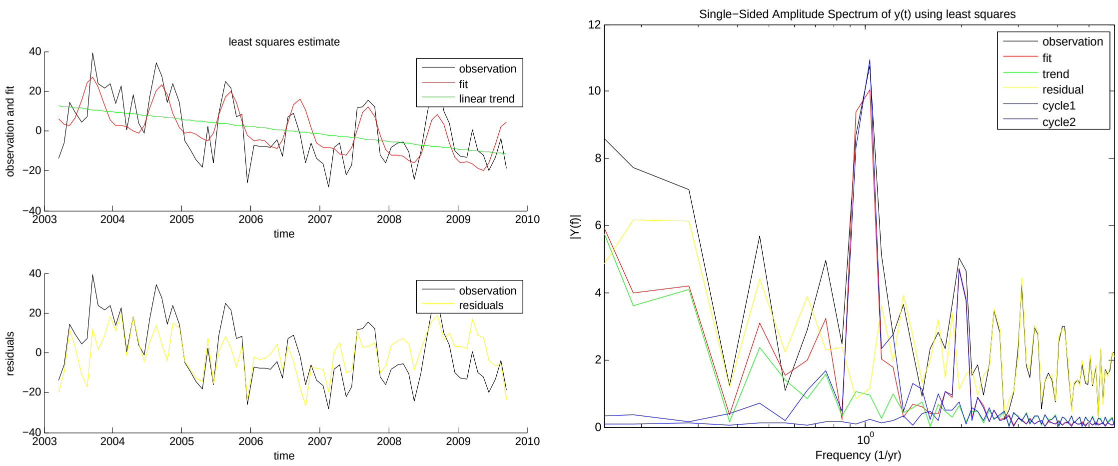
If variances are set to zero, these stay zero during the estimation of disturbance variances in **EM.m**.

2.1.4 Some other settings

- *settings.fixprocvar*: if true, keep disturbance variances to initial values.
- *settings.lsq*: if true, also a least squares fit is determined for comparison. Only part of *run_state_space.m*.
- *plotEM* plot the results of iteration of the EM parameter estimation.
- *settings.filter* filter type, current options 'DurbinKoopman' (default), or 'SquareRoot' a square root filter that assures positive variance estimates, both from Durbin and Koopman (2012).
- *tseries.missingepochs*: in case of equally spaced time, the indexes of *tseries.time* with missing observations can be provided by the user in *tseries.missingepochs*. In that case *tseries.y* should contain NaN for those indexes. *tseries.missingepochs* does not have to be provided by the user, but can be produced automatically by **FindMissingEpochs.m**. To check that missing epochs are properly assigned, set *settings.checkplots* = true. When *settings.continuoustime* is true, *tseries.missingepochs* should be left empty.

3 Example: GRACE time series

Here an example will be given of estimating trend and cycles from a GRACE time series with strong annual and semi-annual periodic signals. To show the improvements of a state space model where we model components stochastically compared to traditional deterministic fitting methods, figure 1 shows the least squares fit to a GRACE time series. The red line shows that the deterministic fit has difficulties to fit all yearly peaks, which hints at either a slowly changing trend or variations in the yearly (cycle) signal. The spectrum in the right panel shows that the yearly cycle fits the peak at a frequency of $\frac{1}{year}$ but leaves considerable signal with frequencies slightly of this peak frequency. The same applies for the semi-annual fit. Besides, the residual contains both signal in the high frequency domain (which we might consider as noise), but also at frequencies longer than a year, which implies that there are long-term variations that deviate from a linear trend.

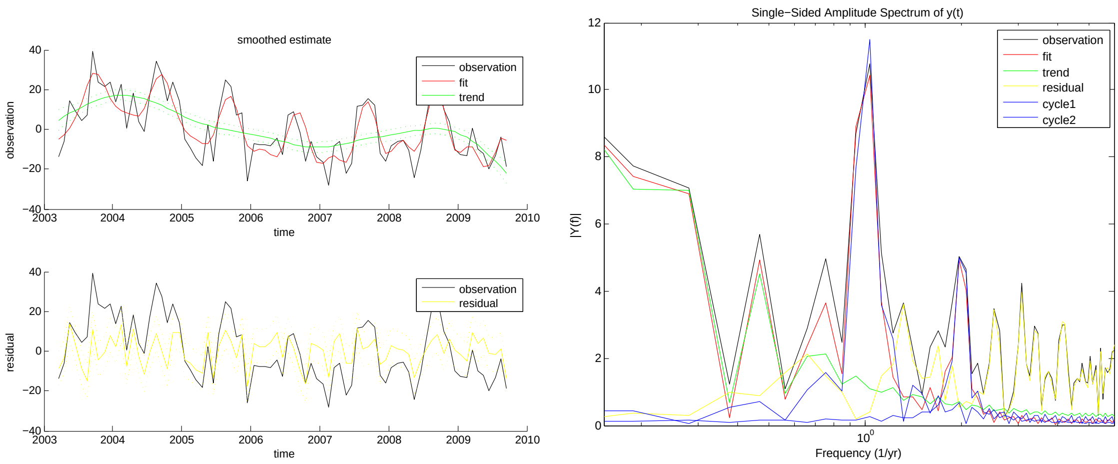


(a) (b)

Figure 1: (a) Least squares fit of linear trend and annual and semi-annual cycles of a GRACE monthly time series; (b) spectrum of GRACE time series and fit.

Now we use the Kalman filter and smoother applying a local linear trend model plus annual and semi-annual cycles. The log likelihood is optimised and the resulting disturbance variances are $\sigma_\epsilon^2 = 74$ for the irregular component and $\sigma_\xi^2 = 0.34$ for the slope (σ_ζ^2 is kept at zero) and $\sigma_{\omega_1}^2 = 0.47; \sigma_{\omega_1^*}^2 = 0.33$ and $\sigma_{\omega_2}^2 = 0.32; \sigma_{\omega_2^*}^2 = 0.39$ for annual and semi-annual cycles respectively. Now the trend is stochastic (local linear trend or integrated random walk as $\zeta = 0$) it can absorb long period changes, that in the deterministic approach ended up in the residuals. Here the residual (or irregular component) mostly contains high frequent signal. Furthermore, the two cycle terms now slightly vary in amplitude and phase throughout the observed period, due to the random walk ω that allows for variations in phase and amplitude.

The variation in cycle terms can be better seen in figure 3 where the individual cycles together with amplitude A and phase ϕ as a function of time are depicted.



(a) (b)

Figure 2: (a) State space fit of stochastic trend and annual and semi-annual cycles of the same GRACE monthly time series as in figure 1, dashed line denotes 1-σ boundaries; (b) spectrum of GRACE time series and fit.

3.1 Combination of stochastic and deterministic components

It is also possible to use the state space approach with mixed deterministic and stochastic components. Components can be made deterministic by setting their respective process noise variance to zero. For example a linear trend can be modeled in combination with stochastic cycle terms. While the EM algorithm can be used to maximise the log likelihood under the condition that $\zeta = \xi = 0$, this likely forces the cycle terms to absorb more long period signal (by increase variances of ω and ω^*). Thus it may be advisable to estimate variances when all components are modeled stochastically and to use those variances for the cycle and irregular terms. Or, when a linear trend is required, a fully stochastic model may be estimated after which by means of post-processing a linear trend is determined from the time variable trend.

References

Durbin, J. and Koopman, S. (2012). *Time series analysis by state space methods*. Oxford University Press.

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press.

Koopman, S. J. (1993). Disturbance smoother for state space models. *Biometrika*, 80(1):117–126.

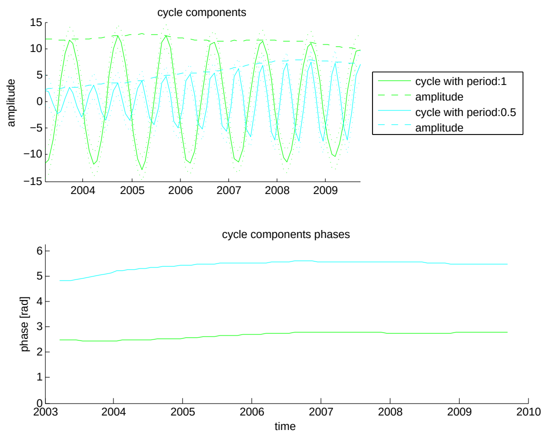


Figure 3: Annual and semi-annual cycle components together with amplitude (upper panel dashed line) and phase ϕ in the lower panel.