BANK OF ENGLAND

CENTRE FOR CENTRAL BANKING STUDIES

ECONOMIC MODELLING AND FORECASTING

A short-term inflation forecast for the UAE: Part II

by

Andy Blake and Ole Rummel
Centre for Central Banking Studies
Bank of England

6 February 2015

1 Using alternative STIF models: the structural time-series model

ARIMA models are not the only way to produce (short-run) inflation forecasts. The most 'complete' or extensive model, in the sense that it nests many of the models we have been working with in Part I, is the structural times-series model (STSM) or the unobserved components (UC) model. Indeed, most of the empirical short-term inflation forecasting models employed by the Bank of England fall into this category.[1] These models form part of the wider state-space analysis, in which time-series observations are assumed to depend linearly on a **state vector** that is unobserved and generated in a dynamic system by a stochastically time-varying process.

The STSM or UC modelling approach in question is generally associated with Harvey (1989). In such a structural time-series or unobserved components model, observations are regarded as made up of distinct **unobserved** components – or building blocks – such as trend, seasonal, regression elements and disturbance terms, each of which is modelled separately.[2] In the current exercise, separate UC models are built for each of the disaggregated CPI indices, and the individual models for the unobserved components are then put together to form a single model. As such, a univariate STSM or UC model of a variable, y_t, has the following generic additive form:

$$y_t = \mu_t + \gamma_t + \psi_t + v_t + \varepsilon_t \qquad \varepsilon_t \sim \text{iid N}(0, \sigma_\varepsilon^2) \tag{1}$$

which models the dependent time-series variable, y_t, as the sum of:

$$y_t = \text{trend}_t + \text{seasonal}_t + \text{cycle}_t + \text{autoregressive}_t + \text{irregular}_t$$

Note the absence of any 'economic' variables in equation (1). Each component of equation (1) is modelled in state-space form and estimated using the Kalman filter.

For example, a slowly varying unobserved **trend** component is generally modelled by the so-called local linear trend (LLT) model, due to Harvey and Todd (1983):

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t \qquad \eta_t \sim \text{iid N}(0, \sigma_\eta^2) \tag{2}$$

$$\beta_t = \beta_{t-1} + \zeta_t \qquad \zeta_t \sim \text{iid N}(0, \sigma_\zeta^2) \tag{3}$$

This local linear trend model, which allows the trend level (μ_t) and slope (β_t) to vary over time, is the standard state-space representation for handling strongly trending series. This model may be useful in cases where a single linear trend does not fit the data well. In essence, the trend is modelled as a random walk with drift. The level component, μ_t, can be regarded as the equivalent of the intercept in the classical linear regression model, $y_t = \mu + \varepsilon_t$, which is obtained by setting all the level disturbances, η_t, in equation (2) equal to zero and with $\mu = \mu_1$, where μ_1 is the first observation of μ in the sample. The key difference is that the intercept μ in a regression model is fixed whereas the level component, μ_t, in equation (2) is allowed to change from time point to time point. The local linear trend model contains a

[1] Around three-quarters of the Bank of England's 30-odd CPI component forecasts come from a univariate STSM or an unobserved components model.
[2] It is the responsibility of the researcher to decide what components are required in a specific situation and then to consider whether they apply to the time series under consideration.

stochastic slope, β_t, which also follows a random walk. The slope component, β_t, can be interpreted as the equivalent of the regression coefficient on a time trend in the classical regression model, where the observed time series, y_t, is regressed on the independent variable time, t, such that $y_t = \mu + \beta t + \varepsilon_t$, with $\mu = \mu_1$ and $\beta = \beta_1$, where β_1 is the initial observation. Again, the important difference is that the regression coefficient or weight, β, is fixed in the classical linear regression model, whereas the slope β_t in the local linear trend model is allowed to change over time.

The most basic representation of the STSM or UC model, is, abstracting from the possible existence of a seasonal, cyclical or autoregressive term, simply the LLT model:

$$y_t = \mu_t + \varepsilon_t$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t$$

$$\beta_t = \beta_{t-1} + \zeta_t$$

where the irregular, level and slope disturbances, ε_t, η_t and ζ_t respectively, are mutually independent, i.e., uncorrelated. In this model, β_t is the local trend rate and μ_t is the local trend itself, which is the local mean value for the observable y_t. In other words, μ_t and β_t are the (time-varying) level and slope of the trend respectively. The stochastic slope parameter, β_t, allows the trend to change smoothly over time.

What we are doing in this case is essentially decomposing the single observable variable, y_t, into three components. According to the model, we should be able to separate them based upon their persistence. Changes to the trend rate, η_t, have the greatest effect on the evolution of the series, while the measurement-equation error, ε_t, has the least effect and the level shock, ζ_t, is in between. With many series, it is rather hard to separate the level shock from the measurement error. In particular, if the variance of η_t is high (relative to the measurement error), the 'trend' can move around quite a bit, possibly coming very close to tracking the data wherever it goes.

In terms of the other unobserved components, the **seasonal** component, if present, can be modelled in one of two ways. The first approach models the unobserved seasonal component, γ_t, by stochastic seasonal dummies. In the case of quarterly data, adding three quarterly stochastic seasonal dummies (γ_{1t}, γ_{2t}, γ_{3t}) to the local linear trend model results in the following functional form with one observation or measurement equation and five state or transition equations:

$$y_t = \mu_t + \gamma_{1t} + \varepsilon_t \qquad \varepsilon_t \sim \text{iid N}(0, \sigma_\varepsilon^2) \tag{4}$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t \qquad \eta_t \sim \text{iid N}(0, \sigma_\eta^2) \tag{5}$$

$$\beta_t = \beta_{t-1} + \zeta_t \qquad \zeta_t \sim \text{iid N}(0, \sigma_\zeta^2) \tag{6}$$

$$\gamma_{1t} = -\gamma_{1,t-1} - \gamma_{2,t-1} - \gamma_{3,t-1} + \omega_t \qquad \omega_t \sim \text{iid N}(0, \sigma_\omega^2) \tag{7}$$

$$\gamma_{2t} = \gamma_{1,t-1} \tag{8}$$

$$\gamma_{3t} = \gamma_{2,t-1} \tag{9}$$

for $t = 1, \dots, T$, which is a local linear trend and seasonal dummy model for a quarterly time series where the seasonal component is allowed to change over time. Note that the sum over the seasonal factors over a full year is approximately zero (equation (7)):

$$\gamma_t = -\sum_{j=1}^{s-1} \gamma_{t-j} + \omega_t \tag{10}$$

where s is the number of seasons in a period, such as a year, and ω_t is white noise. The three white-noise processes η_t, ζ_t and ω_t are assumed to be independent. [3]

As discussed in Proietti's (2000) review of different seasonal specifications and their properties in state-space models, the seasonal dummy model is not the only approach to incorporate time-varying seasonal effects in UC time-series models.

An alternative approach models the unobserved **seasonal** component as a periodic component of fixed period. The number of seasonal frequencies in a period, such as a year, is again given by integer s. When s is even, $[s/2] = s/2$, and when s is odd, $[s/2] = (s - 1)/2$. The trigonometric seasonal form is given by:

$$\gamma_t = \sum_{j=1}^{[s/2]} \gamma_{jt} \tag{11}$$

where each γ_{jt} is generated by:

$$\begin{bmatrix} \gamma_{jt} \\ \gamma^*_{jt} \end{bmatrix} = \begin{bmatrix} \cos(\lambda_j) & \sin(\lambda_j) \\ -\sin(\lambda_j) & \cos(\lambda_j) \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1} \\ \gamma^*_{j,t-1} \end{bmatrix} + \begin{bmatrix} \omega_{jt} \\ \omega^*_t \end{bmatrix} \qquad j = 1, \dots, [s/2] \quad t = 1, \dots, T \tag{12}$$

where $\lambda_j = 2\pi j/s$ is the frequency (in radians) and the seasonal disturbances ω_t and ω_t* are two mutually uncorrelated, normally and independently distributed disturbances with zero mean and common variance, σ_ω^2. Note that, according to equation (11), more than one seasonal term will enter into the unobserved components model. In fact, there will be two seasonal terms with quarterly data (j = 1, ... , [s/2] = 1, ... , [4/2] = 1, 2) and six seasonal terms with monthly data. For s even, the final component at $j = s/2$, equivalent to $\lambda_{(s/2)} = 2\pi(s/2)/s = \pi$, collapses to:

$$\gamma_{jt} = \gamma_{j,t-1}\cos(\lambda_j) + \omega_{jt} \tag{13}$$

The unobserved **cycle** component, following Harvey and Jaeger (1993), is modelled trigonometrically in a similar way to the seasonal component. Specifically, it consists of one or more cycles defined by the pair of equations:[4]

$$\begin{bmatrix} \psi_t \\ \psi^*_t \end{bmatrix} = \rho_\psi \begin{bmatrix} \cos(\lambda_c) & \sin(\lambda_c) \\ -\sin(\lambda_c) & \cos(\lambda_c) \end{bmatrix} \begin{bmatrix} \psi_{t-1} \\ \psi^*_{t-1} \end{bmatrix} + \begin{bmatrix} \kappa_t \\ \kappa^*_t \end{bmatrix} \qquad t = 1, \dots, T \tag{14}$$

[3] Appendix A explains how to set up the LLT and seasonal dummy model with monthly data.
[4] Appendix D illustrates EViews code that shows how the cyclical component is set up in EViews.

where $0 < \rho_\psi \leq 1$ is the damping factor and $0 \leq \lambda_c \leq \pi$, where λ_c is the cycle periodicity in radians (alternatively, $2\pi/\lambda_c$ is the period of the cycle). When $\rho_\psi = 1$, the cycle reduces to a deterministic sine-cosine wave, although the component is still stochastic since the initial values ψ_1 and ψ_1^* are stochastic variables with mean zero and variance σ_ψ^2. In addition, κ_t, $\kappa_t^* \sim N(0,\sigma_\kappa^2)$, i.e., the two processes share the same variance. The model with stochastic trend, μ_t; seasonal effect at time t, γ_t; and irregular component, ε_t, is – following Harvey (1989) – referred to as the basic structural model.

The unobserved **autoregressive** component has the following straightforward representation:

$$v_t = \rho_v v_{t-1} + \xi_t \qquad \xi_t \sim \text{iid } N(0, \sigma_\xi^2) \tag{15}$$

where $0 < \rho_v < 1$, while the unobserved **irregular** component, ε_t, with variance-covariance matrix, Σ_ε, is not explained in the model.

In the short-term inflation forecast carried out by the Bank of England, the exact specification may vary by disaggregated CPI index series and is revised on an *ad hoc* basis. In particular, we allow the trend specification to vary across disaggregated CPI indices. Importantly, the STSMs or UC models are re-estimated each month.

For illustrative purposes, we will build unobserved components models for the `food_soft_drinks`, `furniture_household`, `housing`, `medical_care`, `miscellaneous_goods`, `recreation` and `transportation` price series.

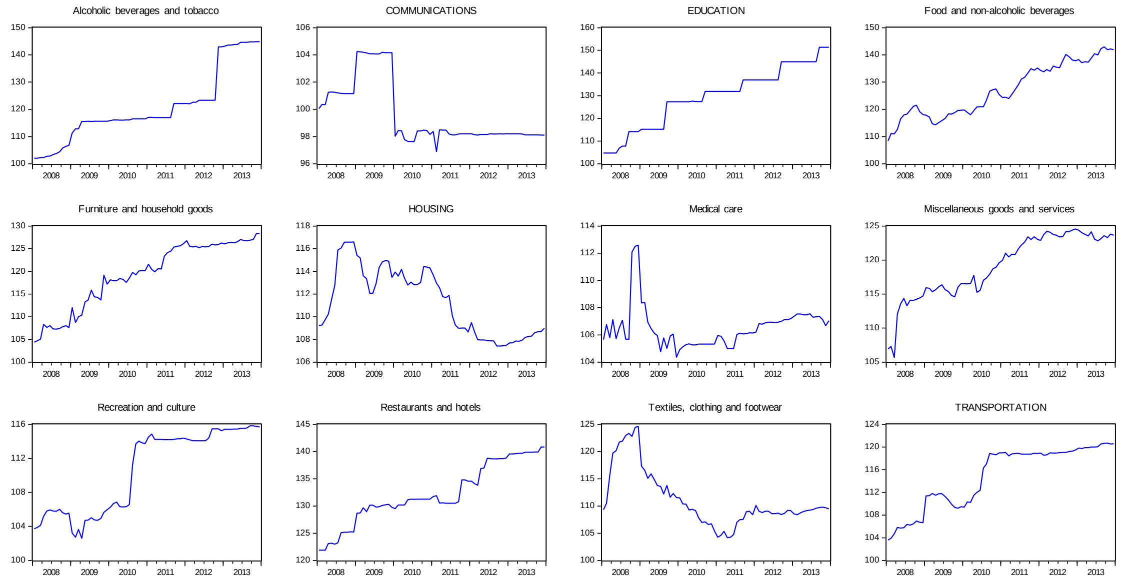
2 Data and EViews programme

All the underlying files can be found in a folder called **STIF – UAE** in the current course **Data** folder.

The STIF used in this exercise is based on a monthly dataset of twelve disaggregated UAE CPI components, spanning the period from January 2008 (`2008M01` in EViews notation) to June 2014 (`2014M06`) for a total of 78 monthly observations. For the purposes of the exercise, we will assume that we have data up to December 2013, i.e., `2013M12`. The forecasts will then be constructed for the six-month period from January 2014 to June 2014 and compared to the actual inflation data over the same period.

Open the workfile **stif_uae.wf1** and look at the group **group_all**. These are the twelve main components of the UAE's composite CPI index. As is quite customary in CPI indices in many countries, the three components with the largest weights are housing (39.33 per cent); food and non-alcoholic beverages (13.94 per cent) and transportation (9.94 per cent). Altogether, these three components account for slightly more than 63 per cent of the UAE composite CPI index. The three components with the lowest weights are recreation and culture (3.07 per cent); medical care (1.12 per cent) and alcoholic beverages and tobacco (0.22 per cent). The time series of the price levels for the twelve components are given in Figure 1.

Figure 1: Disaggregated UAE composite CPI components
(January 2008 – December 2013)



Many of the series show some form of (upward) trend over the time period (or at least large parts of it), with some displaying additional idiosyncratic features (such as `medical_care`, `housing` and `communications`). Amongst other things, this raises the question of what time period to use for the estimation of the underlying models. For example, would you want to use the entire sample period for `transportation`? In a similar vein, `education` looks quite deterministic, with regular increases in August, and we will return to this point below. `Food_soft_drinks`, on the other hand, is clearly upward sloping, but there are substantial short-term movements around the trend. Without further analysis, we cannot say whether these movements are seasonal or not. Based on visual inspection alone, there are – apart from February 2011 for `communications` and the period of 2008 Q4 for `medical_care` – no other obvious outliers (aberrant observations) that need to be accounted for. Even so, we may want to think about the step-ups (level interventions) in `beverages_tobacco` (September 2011 and November 2012) and `recreation_culture` (September 2009 as well as September 2012) a bit more.

3 Estimation

Open the programme **ss_uae_component.prg**, which contains prototype STSMs for `food_soft_drinks`, `furniture_household`, `housing`, `medical_care`, `miscellaneous_goods`, `recreation` and `transportation`. Note that this group includes the `housing`, `food_soft_drinks` and `transportation` components, which have the three biggest weights in the UAE composite CPI index. At the same time, these are prototype models in that they are initial attempts (arrived at by trial and error) to model the seven underlying series. We make no claim that these models will be good models for either data-description or forecasting. The prototype STSMs in EViews are given by:[5]

[5] The different EViews codes that we will encounter in this exercise are discussed in more detail in Appendix B.

```
smpl 2008m01 2013m12

sspace ss_food

ss_food.append @signal food_soft_drinks = mu
ss_food.append @state mu = mu(-1) + beta(-1) + [var=exp(c(2))]
ss_food.append @state beta = beta(-1) + [var=exp(c(3))]

ss_food.ml


sspace ss_furniture

ss_furniture.append furniture_household = mu + [var=exp(c(4))]
ss_furniture.append @state mu = mu(-1) + beta(-1)
ss_furniture.append @state beta = beta(-1)

ss_furniture.ml


sspace ss_housing

ss_housing.append housing = mu + sv1 + [var=exp(c(7))]
ss_housing.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(8))]
ss_housing.append @state beta = beta(-1) + [var=exp(c(9))]
ss_housing.append @state sv1 = c(10)*sv1(-1) +
[var=exp(c(11))]

ss_housing.ml


sspace ss_medical

ss_medical.append medical_care = mu + sv1
ss_medical.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(13))]
ss_medical.append @state beta = beta(-1)
ss_medical.append @state sv1 = c(15)*sv1(-1) + [var=1]

ss_medical.ml


sspace ss_misc

ss_misc.append miscellaneous_goods = mu + sv1 +
[var=exp(c(17))]
```



```
ss_misc.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(18))]
ss_misc.append @state beta = beta(-1) + [var=exp(c(19))]
ss_misc.append @state sv1 = c(20)*sv1(-1) + [var=exp(c(21))]

ss_misc.ml


sspace ss_recreation

ss_recreation.append recreation_culture = mu +
[var=exp(c(21))]
ss_recreation.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(22))]
ss_recreation.append @state beta = beta(-1)

ss_recreation.ml


sspace ss_trans

ss_trans.append transportation = mu
ss_trans.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(25))]
ss_trans.append @state beta = beta(-1)

ss_trans.ml
```

Now run the programme **ss_uae_component.prg** by clicking on **Run** on the programme window. You will find that EViews has now created – and saved – the seven new estimated state-space objects `ss_food`, `ss_furniture`, `ss_housing`, `ss_medical`, `ss_misc`, `ss_recreation` and `ss_trans` in the workfile.

4 Forecasting with the state-space model

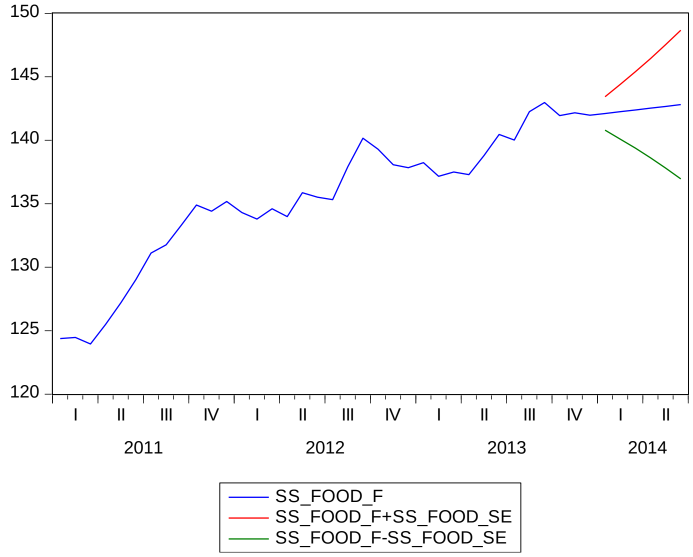
In the next step, we will use these seven state-space objects and produce a forecast over the period from January to June 2014. Forecasting with state-space models is very similar to the forecasting procedures employed for a standard model. Open the programme **ss_uae_component_forecast.prg**. When run (again, click on **Run** in the programme window), this automatically forecasts the signal series for both models over the period from `2014M01` to `2014M06` and graphs the outputs:

```
' Do forecasts

smpl 2014m01 2014m06

ss_trans.forecast @signal ss_trans_f @signalse ss_trans_se
```

```
group sstrans ss_trans_f (ss_trans_f+ss_trans_se) (ss_trans_f-
ss_trans_se)

ss_furniture.forecast @signal ss_furn_f @signalse ss_furn_se
group ssfurn ss_furn_f (ss_furn_f+ss_furn_se) (ss_furn_f-
ss_furn_se)

ss_food.forecast @signal ss_food_f @signalse ss_food_se
group ssfood ss_food_f (ss_food_f+ss_food_se) (ss_food_f-
ss_food_se)

ss_housing.forecast @signal ss_hous_f @signalse ss_hous_se
group sshous ss_hous_f (ss_hous_f+ss_hous_se) (ss_hous_f-
ss_hous_se)

ss_medical.forecast @signal ss_med_f @signalse ss_med_se
group ssmed ss_med_f (ss_med_f+ss_med_se) (ss_med_f-ss_med_se)

ss_misc.forecast @signal ss_misc_f @signalse ss_misc_se
group ssmisc ss_misc_f (ss_misc_f+ss_misc_se) (ss_misc_f-
ss_misc_se)

ss_recreation.forecast @signal ss_rec_f @signalse ss_rec_se
group ssrec ss_rec_f (ss_rec_f+ss_rec_se) (ss_rec_f-ss_rec_se)

' Put in back data
smpl 2009m01 2013m12
genr ss_trans_f = transportation
genr ss_furn_f = furniture_household
genr ss_food_f = food_soft_drinks
genr ss_hous_f = housing
genr ss_med_f = medical_care
genr ss_misc_f = miscellaneous_goods
genr ss_rec_f = recreation_culture

' Adjust sample period and plot
smpl 2011m01 2014m06
sstrans.line
ssfurn.line
ssfood.line
sshous.line
ssmed.line
ssmisc.line
ssrec.line
```

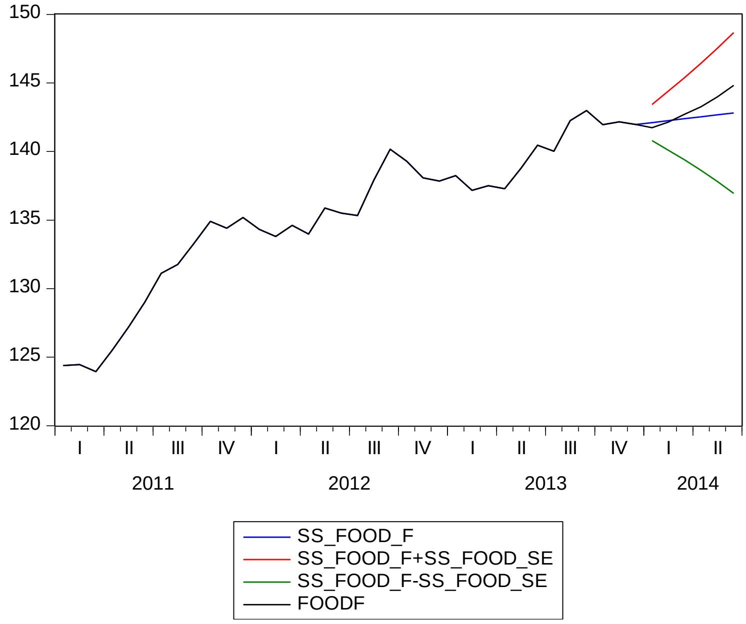
An example of one of these graphs (for `food_soft_drinks`), showing the mean forecast (denoted `ss_food_f`) as well as the plus and minus one standard error forecast interval (given by (ss_food_f+ss_food_se) and (ss_food_f-ss_food_se) respectively), is shown in Figure 2.

Figure 2: Forecast of the `food_soft_drinks` state-space model with one standard-error bounds (January 2014 – June 2014)



The difference between the forecasts based on the estimated state-space model (ss_food_f) and the AR model estimated by OLS in Part I of this exercise (foodf) is illustrated in Figure 3.

Figure 3: Comparison between the AR and state-space model forecasts for the `food_soft_drinks` inflation component (January 2014 – June 2014)



We can see that in contrast to the autoregressive forecast for the level of the `food_soft_drinks` price index from Part I of the exercise, which was upward sloping after the second month of the forecast period, the STSM forecast continues the gentle increase seen prior to the start of the forecast period in January 2014. An interesting picture emerges for the `furniture_household` CPI component (not shown), which includes a step-up in January 2014 which is absent from the simple AR forecast. For the `housing` CPI component (not shown), we find that the forecasts using the state-space model are slightly steeper than the corresponding forecasts using an AR model. The opposite is the case for the `medical_care` CPI component (not shown), where the simple AR forecast shows a steeper fall over the forecast period than the corresponding state-space model forecast. Along similar lines, the state-space model forecast for the `miscellaneous_goods` CPI component (not shown) displays an extended fall and levelling-off in the index, whereas the simple AR forecast shows an increase over the forecast period. The reverse is true for the state-space model forecast for both the `recreation_culture` and the `transportation` CPI components (not shown): the simple AR forecasts display a levelling-off in the index, whereas the state-space model forecasts shows steep increases over the forecast period.

5 CPI forecast

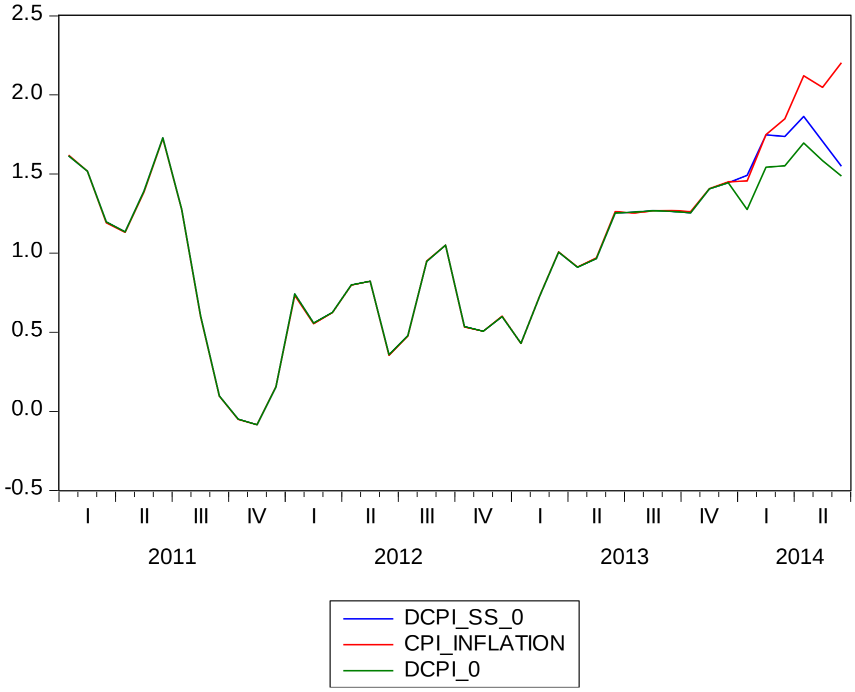
The forecast is once again straightforward: we replace the seven AR models in the CPI index by the seven state-space models from the previous Section and simply weight together the seven state-space forecasts with the remaining five AR forecasts, using the original CPI weights as a model of the all-items CPI inflation index. The programme **ss_uae_cpi_forecast.prg** defines the weights, specifies a state-space model called **stif_ss**, solves the model to produce the forecasts and then calculates and plots the inflation rate:

```
vector(12) weights
weights.fill 13.936, 0.218, 7.580, 39.334, 4.208, 1.124,
9.941, 6.932, 3.067, 4.004, 4.348, 5.308

model stif_ss

stif_ss.append composite_ss = (weights(1)*ss_food_f +
weights(2)*bevsf + weights(3)*textf + weights(4)*ss_hous_f +
weights(5)*ss_furn_f + weights(6)*ss_med_f +
weights(7)*ss_trans_f + weights(8)*commsf +
weights(9)*ss_rec_f + weights(10)*eduf + weights(11)*restf +
weights(12)*ss_misc_f)/100

smpl 2010m01 2014m06
stif_ss.solve

smpl 2011m01 2014m06
genr dcpi_ss_0 = 100*(composite_ss_0-composite_ss_0(-
12))/composite_ss_0(-12)
```

```
group inflation_ss dcpi_ss_0 cpi_inflation dcpi_0
inflation_ss.line
```

Have a look at the programme and the way it creates the model and then run it by clicking on **Run** in the programme window. The output, shown in Figure 4, consists of the three series `cpi_inflation`, which is the rate of inflation based on the actual underlying UAE data; `dcpi_0`, which is the disaggregated inflation forecast based on the simple AR models only; and `dcpi_ss_0`, which is the disaggregated inflation forecast containing both AR and state-space model forecasts. Is the forecast any good?

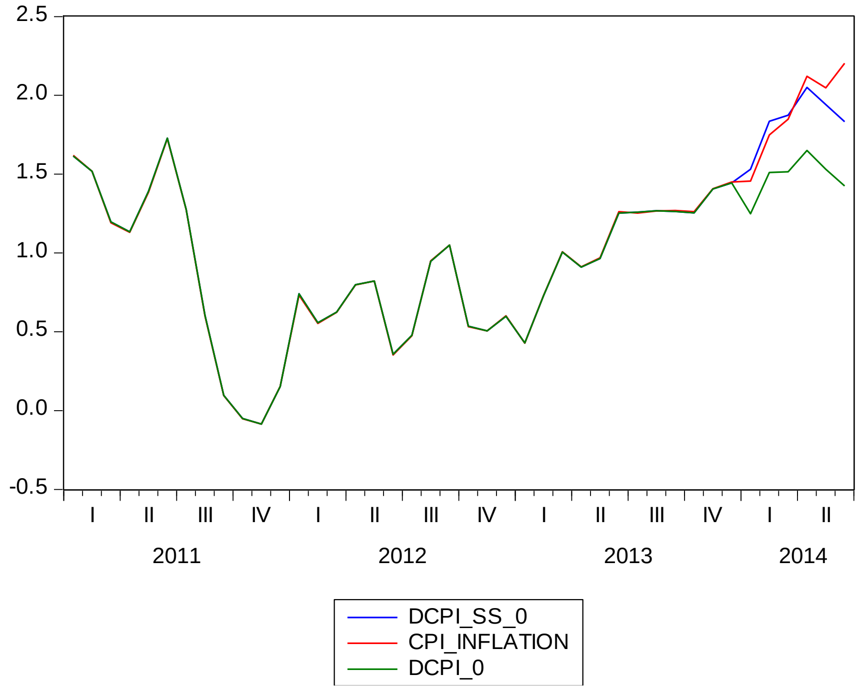
Figure 4: Forecast of the composite CPI inflation rate (January 2014 – June 2014)



After the inclusion of the seven state-space models, the blue line of the new forecast has moved much closer to the red line representing the actual data for the first two months of the forecast period. While the performance of the forecast invariably deteriorates as we move further into the forecast period, the short-term forecast performance of the augmented model is quite decent – and definitely better than that of the simple AR forecasting models .

We should not forget that these prototype models constitute by no means the state-of-the-art of short-term forecasting models. Rather annoyingly, I failed to record the exact specifications of the seven STSMs that resulted in the forecast shown in Figure 5.

Figure 5: Forecast of the composite CPI inflation rate (January 2014 – June 2014)



Using these seven unrecorded state-space models translated into the blue line of the new forecast moving much closer to the red line representing the actual data for four months of the forecast period. While the performance of the forecast once again deteriorates as we move further into the forecast period, the short-term performance is quite decent. In fact, at the four-month-ahead horizon, the performance of the STIF is more or less spot on, with the difference between the actual and the forecasted value amounting to the second significant digit.

References and further reading

Box, G E P and Tiao, G C (1975), 'Intervention analysis with applications to economic and environmental problems', *Journal of the American Statistical Association*, Vol. 70, No. 349, pages 70-79.

De Jong, P and Penzer, J (1998), 'Diagnosing shocks in time series', *Journal of the American Statistical Association*, Vol. 93, No. 442, pages 796-806.

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

Harvey, A C and Jaeger, A (1993), 'Detrending, stylised facts and the business cycle', *Journal of Applied Econometrics*, Vol. 8, No. 3, pages 231-47.

Harvey, A C and Koopman, S J (1992), 'Diagnostic checking of unobserved-components time series models', *Journal of Business & Economic Statistics*, Vol. 10, No. 4, pages 377-89.

Harvey, A C and Todd, P H J. (1983), 'Forecasting economic time series with structural and Box-Jenkins models: a case study', *Journal of Business & Economic Statistics*, Vol. 1, No. 4, pages 299-315.

Proietti, T (2000), 'Comparing seasonal components for structural time series models', *International Journal of Forecasting*, Vol. 16, No. 2, pages 247-60.

van den Bossche, F A M (2011), 'Fitting state space models with EViews', *Journal of Statistical Software*, Vol. 41, No. 8. http://www.jstatsoft.org/v41/i08/paper.

Appendix A: Monthly seasonal dummies

The general linear Gaussian state-space model for a time series of T observations, given by $y_1, \ldots, y_T$, can be formulated as:

$$y_t = Z_t\alpha_t + \varepsilon_t \qquad \varepsilon_t \sim \text{iid } N(0, H_t) \tag{A.1}$$

$$\alpha_t = T_t\alpha_{t-1} + R_t\chi_t \qquad \chi_t \sim \text{iid } N(0, Q_t) \tag{A.2}$$

for $t = 1, \ldots , T$. In equations (A.1) and (A.2), α_t is the state vector, ε_t and χ_t are disturbance vectors and the system matrices Z_t, T_t, R_t, H_t and Q_t are fixed and known, even though a selection of elements may depend upon an unknown parameter vector. The (n×1) observation vector, y_t, contains the T observations at time t and the (m×1) state vector α_t is unobserved. The (n×1) irregular vector, ε_t, has mean zero and (n×n) variance matrix, H_t. The (n×m) matrix Z_t links the observation vector, y_t, with the unobservable state vector, α_t, and may consist of regression variables. The (m×m) transition matrix T_t in equation (A.2) determines the dynamic evolution of the state vector. The (r×1) disturbance vector, χ_t, for the state vector has zero mean and (r×r) variance matrix, Q_t. The observation and transition disturbances ε_t and χ_t are assumed to be serially independent and independent of each other at all time points. In many standard cases, $r = m$ and matrix R_t is the identity matrix I_m. In other cases, matrix R_t is an (m×r) selection matrix with $r < m$. Although matrix R_t can be specified freely, it is often composed of a selection from the first r columns of the identity matrix I_m.

Section 2 introduced the specification of the stochastic seasonal dummy component for quarterly data. In the case of a monthly time series, the local linear trend and seasonal dummy model using equations (A.1) and (A.2) will be given by:

$$\alpha_t = \begin{pmatrix} \mu_t \\ \beta_t \\ \gamma_{1t} \\ \gamma_{2t} \\ \gamma_{3t} \\ \gamma_{4t} \\ \gamma_{5t} \\ \gamma_{6t} \\ \gamma_{7t} \\ \gamma_{8t} \\ \gamma_{9t} \\ \gamma_{10t} \\ \gamma_{11t} \end{pmatrix}, \quad \chi_t = \begin{pmatrix} \eta_t \\ \zeta_t \\ \omega_t \end{pmatrix}, \quad T_t = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$Z_t = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$H_t = \sigma_\varepsilon^2,$$

$$Q_t = \begin{pmatrix} \sigma_\xi^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\zeta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad R_t = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The monthly state-space model written out will have one observation or measurement equation and 13 state or transition equations:

$$y_t = \mu_t + \gamma_{1t} + \varepsilon_t \tag{A.3}$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t \tag{A.4}$$

$$\beta_t = \beta_{t-1} + \zeta_t \tag{A.5}$$

$$\gamma_{1t} = -\gamma_{1,t-1} - \gamma_{2,t-1} - \gamma_{3,t-1} - \gamma_{4,t-1} - \gamma_{5,t-1} - \gamma_{6,t-1} - \gamma_{7,t-1} - \gamma_{8,t-1} - \gamma_{9,t-1} - \gamma_{10,t-1} - \gamma_{11,t-1} + \omega_t \tag{A.6}$$

$$\gamma_{2t} = \gamma_{1,t-1} \tag{A.7}$$

$$\gamma_{3t} = \gamma_{2,t-1} \tag{A.8}$$

$$\gamma_{4t} = \gamma_{3,t-1} \tag{A.9}$$

$$\gamma_{5t} = \gamma_{4,t-1} \tag{A.10}$$

$$\gamma_{6t} = \gamma_{5,t-1} \tag{A.11}$$

$$\gamma_{7t} = \gamma_{6,t-1} \tag{A.12}$$

$$\gamma_{8t} = \gamma_{7,t-1} \tag{A.13}$$

$$\gamma_{9t} = \gamma_{8,t-1} \tag{A.14}$$

$$\gamma_{10t} = \gamma_{9,t-1} \tag{A.15}$$

$$\gamma_{11t} = \gamma_{10,t-1} \tag{A.16}$$

Appendix B: EViews code for the UAE short-term inflation forecast

The seven representative STSMs are estimated using the EViews programme called **ss_uae_component.prg**, which will be described below.

```
smpl 2008m01 2013m12

sspace ss_food

ss_food.append @signal food_soft_drinks = mu
ss_food.append @state mu = mu(-1) + beta(-1) + [var=exp(c(2))]
ss_food.append @state beta = beta(-1) + [var=exp(c(3))]

ss_food.ml
```

The first line of the programme sets a common sample period from January 2008 to December 2013. The second line defines and creates an EViews state-space object for the `food_soft_drinks` price index series called `ss_food`. The next three lines define equations (1), (2) and (3) for this particular state-space model and append them to the state-space object. We have a measurement, signal or observation equation for `food_soft_drinks` and a state or transition equation each for `mu` and `beta`. The observation or measurement equation consists of a time-varying trend (`mu`) that is observed without (measurement) error, which is modelled by a straightforward LLT model. We note that both the state equations for `mu` and `beta` include an error term. Note also that we have dispensed with any starting values for the maximum-likelihood observation, which would have been done with the `param` command and starting values for `c(2)` and `c(3)`.[6] The final line instructs EViews to estimate the state-space object called `ss_food` using maximum likelihood (ML).

```
sspace ss_furniture

ss_furniture.append furniture_household = mu + [var=exp(c(4))]
ss_furniture.append @state mu = mu(-1) + beta(-1)
ss_furniture.append @state beta = beta(-1)

ss_furniture.ml
```

The second part of the programme estimates the `furniture_household` STSM. The first line opens the EViews state-space object called `ss_furniture`. As above, the following three lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. We have a measurement, signal or observation equation for `furniture_household` and a state or transition equation each for `mu` and `beta`. Note that we can drop the `@signal` command when defining the measurement or observation equation. In contrast to the `ss_food` state-space object, the measurement equation now

[6] The `param` statement for a particular state-space model can go anywhere in the programme and does not necessarily have to be either at the beginning or the end of the code.

includes an error term, while the two transition equations do not. The final line instructs EViews to estimate the state-space object called `ss_furniture` using maximum likelihood (ML).

```
sspace ss_housing

ss_housing.append housing = mu + sv1 + [var=exp(c(7))]
ss_housing.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(8))]
ss_housing.append @state beta = beta(-1) + [var=exp(c(9))]
ss_housing.append @state sv1 = c(10)*sv1(-1) +
[var=exp(c(11))]

ss_housing.ml
```

The third part of the programme estimates the `housing` STSM. The first line opens the EViews state-space object called `ss_housing`. The following four lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. The measurement, signal or observation equation for `housing` now also includes an unobserved autoregressive component, denoted `sv1`, meaning that we need an additional state equation for this new unobserved component. Altogether, we now have three transition equations, i.e., one each for `mu`, `b` and `sv1`. Each of these four equations includes an error term. The final line instructs EViews to estimate the state-space object called `ss_housing` using maximum likelihood (ML).

```
sspace ss_medical

ss_medical.append medical_care = mu + sv1
ss_medical.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(13))]
ss_medical.append @state beta = beta(-1)
ss_medical.append @state sv1 = c(15)*sv1(-1) + [var=1]

ss_medical.ml
```

The next part of the programme estimates the `medical_care` STSM. The first line opens the EViews state-space object called `ss_medical`. The following four lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. The measurement, signal or observation equation for `medical_care` includes an unobserved autoregressive component, denoted `sv1`, but no measurement error term. In the state or transition equation for `sv1`, the variance of the associated error term has been set to 1. With an estimated value of -1.78×10^{-6}, the variance of the autoregressive term is equal to $\mathrm{Var}(v_t) = \sigma_v^2 = e^{-1.78\times10^{-6}} = 1$ (including the appropriate specification to estimate this coefficient is left as an optional exercise). Only two of the four equations include an error term. The final line instructs EViews to estimate the state-space object called `ss_medical` using maximum likelihood (ML).

```
sspace ss_misc

ss_misc.append miscellaneous_goods = mu + sv1 +
[var=exp(c(17))]
ss_misc.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(18))]
ss_misc.append @state beta = beta(-1) + [var=exp(c(19))]
ss_misc.append @state sv1 = c(20)*sv1(-1) + [var=exp(c(21))]

ss_misc.ml
```

This part of the programme estimates the `miscellaneous_goods` STSM. The first line opens the EViews state-space object called `ss_misc`. The following four lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. Once again, the measurement, signal or observation equation for `miscellaneous_goods` includes an unobserved autoregressive component, denoted `sv1`. As we can see, all four equations include an error term. The final line instructs EViews to estimate the state-space object called `ss_misc` using maximum likelihood (ML).

```
sspace ss_recreation

ss_recreation.append recreation_culture = mu +
[var=exp(c(22))]
ss_recreation.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(23))]
ss_recreation.append @state beta = beta(-1)

ss_recreation.ml
```

The next part of the programme estimates the `recreation_culture` STSM. The first line opens the EViews state-space object called `ss_recreation`. The following three lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. The measurement, signal or observation equation for `recreation_culture` includes an unobserved time-varying trend as well as a measurement error, and only two equations include an error term. The final line instructs EViews to estimate the state-space object called `ss_recreation` using maximum likelihood (ML).

```
sspace ss_trans

ss_trans.append transportation = mu
ss_trans.append @state mu = mu(-1) + beta(-1) +
[var=exp(c(25))]
ss_trans.append @state beta = beta(-1)

ss_trans.ml
```

The final part of the programme estimates the transportation STSM. The first line opens the EViews state-space object called ss_trans. The following three lines define the appropriate form of equations (1), (2) and (3) and append them to the state-space object. The measurement, signal or observation equation for transportation includes an unobserved time-varying trend that is observed without measurement error, and only the transition equation for the unobserved trend component includes an error term. The final line instructs EViews to estimate the state-space object called ss_trans using maximum likelihood (ML).

The individual components that we have just estimated are then forecast over the period from January 2014 to June 2014 using the EViews programme called **ss_uae_component_forecast.prg**. The programme is again divided into several steps, which will be described below. All three forecasts generates a mean forecast, which will be labelled by the suffix _f, as well as the forecast standard error, which is labelled by the suffix _se. Using the EViews command group, we then create a group consisting of the mean forecast as well as the forecast confidence intervals, defined as the mean forecast plus and minus one standard error.

```
' Do forecasts

smpl 2014m01 2014m06
```

This simply sets the sample period to the forecast period over which we want to forecast the state-space objects for food_soft_drinks, furniture_household, housing, medical_care, miscellaneous_goods, recreation and transportation.

```
ss_trans.forecast @signal ss_trans_f @signalse ss_trans_se
group sstrans ss_trans_f (ss_trans_f+ss_trans_se) (ss_trans_f-
ss_trans_se)
```

This command takes the state-space object called ss_trans and forecasts it over the period we have set before. It creates the level forecast, denoted by ss_trans_f, as well as the standard error of the level forecast, denoted by ss_trans_se.

```
group sstrans ss_trans_f (ss_trans_f+ss_trans_se) (ss_trans_f-
ss_trans_se)
```

We then create a new group of variables for plotting, consisting of the level forecast (ss_trans_f) as well as the plus and minus one standard-error confidence interval (ss_trans_f+ss_trans_se and ss_trans_f-ss_trans_se respectively).

```
ss_furniture.forecast @signal ss_furn_f @signalse ss_furn_se
group ssfurn ss_furn_f (ss_furn_f+ss_furn_se) (ss_furn_f-
ss_furn_se)

ss_food.forecast @signal ss_food_f @signalse ss_food_se
```

```
group ssfood ss_food_f (ss_food_f+ss_food_se) (ss_food_f-
ss_food_se)

ss_housing.forecast @signal ss_hous_f @signalse ss_hous_se
group sshous ss_hous_f (ss_hous_f+ss_hous_se) (ss_hous_f-
ss_hous_se)

ss_medical.forecast @signal ss_med_f @signalse ss_med_se
group ssmed ss_med_f (ss_med_f+ss_med_se) (ss_med_f-ss_med_se)

ss_misc.forecast @signal ss_misc_f @signalse ss_misc_se
group ssmisc ss_misc_f (ss_misc_f+ss_misc_se) (ss_misc_f-
ss_misc_se)

ss_recreation.forecast @signal ss_rec_f @signalse ss_rec_se
group ssrec ss_rec_f (ss_rec_f+ss_rec_se) (ss_rec_f-ss_rec_se)
```

We repeat the same steps for the state-space objects called `ss_furniture`, `ss_food`, `ss_housing`, `ss_medical`, `ss_misc` and `ss_recreation`.

```
' Put in back data

smpl 2009m01 2013m12

genr ss_trans_f = transportation
genr ss_furn_f = furniture_household
genr ss_food_f = food_soft_drinks
genr ss_hous_f = housing
genr ss_med_f = medical_care
genr ss_misc_f = miscellaneous_goods
genr ss_rec_f = recreation_culture
```

These nine lines take the historical data for `transportation`, `furniture_household`, `food_soft_drinks`, `housing`, `medical_care`, `miscellaneous_goods` and `recreation_culture` over the period from January 2009 to December 2013 and append it before the forecasted values.

```
' Adjust sample period and plot

smpl 2011m01 2014m06

sstrans.line
ssfurn.line
ssfood.line
sshous.line
ssmed.line
ssmisc.line
ssrec.line
```

The final bit of the programme adjusts the sample period to cover some historical data as well as the forecast period. It then plots the variables contained in the seven groups (sstrans, ssfurn, ssfood, sshous, ssmed, ssmisc and ssrec) in seven separate line charts.

The individual forecasts for the seven components that we have just produced are then combined using the EViews programme called **ss_uae_cpi_forecast.prg**. The programme is again divided into several steps, which will be described below. Note that the individual forecasts are combined using the CPI weights. The programme defines the weights, specifies the model and solves for the forecast and then calculates and plots the generated inflation rate forecast (dcpi_ss_0), together with the actual composite CPI inflation rate (cpi_inflation) and the earlier forecast from simple AR models (dcpi_0).

```
vector(12) weights
weights.fill 13.936, 0.218, 7.580, 39.334, 4.208, 1.124,
9.941, 6.932, 3.067, 4.004, 4.348, 5.308
```

We define an empty vector consisting of twelve elements using the EViews command vector(12) and call it weights. We then collect the seven weights of the individual components and fill the vector weights with them. The use of the vector weights will become more obvious below.

```
model stif_ss
```

Forecasting is most easily done in an EViews object called a **model**. The above line creates a new EViews model called stif_ss.

```
stif_ss.append composite_ss = (weights(1)*ss_food_f
+ weights(2)*bevsf + weights(3)*textf + weights(4)*ss_hous_f
+ weights(5)*ss_furn_f + weights(6)*ss_med_f
+ weights(7)*ss_trans_f + weights(8)*commsf
+ weights(9)*ss_rec_f + weights(10)*eduf + weights(11)*restf
+ weights(12)*ss_misc_f)/100
```

The section above generates the model as consisting of the individual forecasts weighted by the respective entries of the weights vector. Note that the seven forecasts for food_soft_drinks, housing, furniture_household, medical_care, miscellaneous_goods, recreation_culture and transportation have been replaced by the alternative forecasts calculated on the basis of estimated state-space models. The remaining five forecasts for beverages_tobacco, textiles_clothing, communications, education and restaurants_hotels still come from the AR models estimated in Part I of this exercise. The advantage of using weights is that we do not have to hard code the weights in the programme. For example, the model could have alternatively been defined by the following command:

```
stif_ss.append composite_ss = (13.936*ss_food_f + 0.218*bevsf
+ 7.580*textf + 39.334*ss_hous_f + 4.208*ss_furn_f
```

```
+ 1.124*ss_med_f + 9.941*ss_trans_f + 6.932*commsf
+ 3.067*ss_rec_f + 4.004*eduf + 4.348*restf
+ 5.308*ss_misc_f)/100
```

But CPI weights are regularly updated. Using the `weights` vector, we only need to update the CPI weights once in the `weights` vector rather than throughout the programme.

```
smpl 2011m01 2014m06
stif_ss.solve
```

The `stif_ss` model is then ‘solved’ over the period from January 2011 to June 2014.

```
smpl 2011m01 2014m06
genr dcpi_ss_0 = 100*(composite_ss_0-composite_ss_0(-
12))/composite_ss_0(-12)
```

This part of the programme calculates the annual inflation rate from January 2011 to June 2014 for the forecasted component CPI index (`dcpi_ss_0`) based on the twelve underlying CPI components, seven of which are STSMs.

```
group inflation_ss dcpi_ss_0 cpi_inflation dcpi_0
inflation_ss.line
```

The final part of the programme groups the series we have just created (`dcpi_ss_0`), the actual inflation rate (`cpi_inflation`) and the earlier inflation forecast using simple AR models only (`dcpi_0`) into a group called `inflation_ss` and plots them.

Appendix C: State-space models with fixed explanatory and intervention variables

Another extension of the basic STSM involves the incorporation of fixed explanatory and intervention variables. The case of the local linear trend model with one regression variable, x_t, and one intervention variable, w_t:

$$y_t = \mu_t + \beta x_t + \lambda w_t + \varepsilon_t \tag{C.1}$$

requires a state vector, α_t, with **four** elements: one for the level component, μ_t; one for the slope component, v_t; one for the regression variable coefficient, β; and one for the intervention variable coefficient, λ. In terms of our state-space model given by equations (A.1) and (A.2), this means that:

$$\alpha_t = \begin{pmatrix} \mu_t \\ v_t \\ \beta_t \\ \lambda_t \end{pmatrix} \quad \eta_t = \begin{pmatrix} \xi_t \\ \zeta_t \end{pmatrix} \quad T_t = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Z_t = \begin{pmatrix} 1 & 0 & x_t & w_t \end{pmatrix}$$

$$H_t = \sigma_\varepsilon^2 \quad Q_t = \begin{pmatrix} \sigma_\xi^2 & 0 & 0 & 0 \\ 0 & \sigma_\zeta^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad R_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

This yields the following state-space model written out in full:

$$y_t = \mu_t + \beta_t x_t + \lambda_t w_t + \varepsilon_t \qquad \varepsilon_t \sim \text{iid } N(0, \sigma_\varepsilon^2) \tag{C.2}$$

$$\mu_t = \mu_{t\text{-}1} + v_t + \xi_t \qquad \xi_t \sim \text{iid } N(0, \sigma_\xi^2) \tag{C.3}$$

$$v_t = v_{t\text{-}1} + \zeta_t \qquad \zeta_t \sim \text{iid } N(0, \sigma_\zeta^2) \tag{C.4}$$

$$\beta_t = \beta_{t\text{-}1} \tag{C.5}$$

$$\lambda_t = \lambda_{t\text{-}1} \tag{C.6}$$

where $\beta = \beta_1 = \beta_t$ and $\lambda = \lambda_1 = \lambda_t$ for $t = 1, \dots, T$. Adding a disturbance term to the state equation (C.5) for β_t would make the regression coefficient effectively a random walk. Doing so would allow therefore for the estimation of time-varying regression effects.

There are different ways of capturing intervention effects, namely a **pulse intervention** to model an outlying observation, a **level intervention** to model a structural break in the level of the series or a **slope intervention** to model a structural break in the slope of the series. If we denote the time point at which an intervention effect occurred by τ, variable w_t can either be represented by a pulse intervention:

$$w_t = \begin{cases} 0, & t < \tau, \quad t > \tau \\ 1, & t = \tau \end{cases} \tag{C.X}$$

or a level intervention:

$$w_t = \begin{cases} 0, & t < \tau \\ 1, & t \geq \tau \end{cases} \tag{C.X}$$

or a slope intervention:

$$w_t = \begin{cases} 0, & t < \tau \\ 1 + t - \tau, & t \geq \tau \end{cases} \tag{C.X}$$

Other types of intervention effects are discussed in Box and Tiao (1975).

Appendix D: EViews code for the unobserved cyclical component

The cyclical component of an STSM for the time series y_t can be estimated in EViews as:

```
param c(1) 0.01 c(2) 0.01 c(3) 0.01 c(4) 0.05 c(5) 0.9 c(6)
0.5

@signal y = mu + psi + [var=exp(c(1))]

@state mu = mu(-1) + beta(-1) + [var=exp(c(2))]
@state beta = beta(-1) + [var=exp(c(3))]
@state psi = c(5)*cos(c(6))*psi(-1) + c(5)*sin(c(6))*psistar(-
             1) + [var=exp(c(4))]
@state psistar = -c(5)*sin(c(6))*psi(-1) +
                  c(5)*cos(c(6))*psistar(-1) + [var=exp(c(4))]
```

