```
// T25preReg3.sce
// np-STEP PREDICTION WITH CONTINUOUS MODEL
// REAL DATA (intensity) from Strahov tunnel are used
// Experiments
// Change: - np  = number of steps of prediction
//         - ord = order of the model for estimation
// Result: - visual comparison of yt and yp
//         - RPE = relative prediction error
// ---------------------------------------------------------------
exec("ScIntro.sce",-1), mode(0)

// PARAMETERS                                                    // 1
np=5;                          // length of prediction (np>=1)  // 2
ord=5;                         // order of the estimated model  // 3
// data selection                                               // 4
k=3;                           // which day is selected         // 5
nz=50;                         // beginning of the day          // 6
nd=288*k;                      // length of the whole one day data // 7
                                                                // 8
// DATA                                                         // 9
dtAll=csvRead('STRAHOV.csv',';');  // load data                 // 10
dt=dtAll((k-1)*288+(nz+1:nd),1);                                // 11
nth=ord+2;                     // size of V                     // 12
```

```
V=1e-8*eye(nth,nth);              // initial information matrix        // 13
thE=rand(nth-1,1,'n');            // prior parametrs                   // 14
yt=dt(1:ord);                     // prior data                        // 15
                                                                       // 16
// TIME LOOP                                                         // 17
for t=ord+1:(nd-np-nz)            // time loop (on-line tasks)         // 18
//  prediction                                                        // 19
  ps=[yt(t-1:-1:t-ord); 1];       // regression vector                 // 20
  yy=ps'*thE;                     // first prediction at t+1           // 21
  for j=1:np                      // predictions for t+2,...,t+np      // 22
    ps=[yy; ps(1:(ord-1)); 1];    // reg. vector (no control)          // 23
    yy=ps'*thE;                   // new prediction (partial)          // 24
  end                                                                  // 25
  yp(t+np,1)=yy;                  // final prediction for time t+np    // 26
                                                                       // 27
  // data measurement (as if)                                        // 28
  yt(t)=dt(t);                    // measuring of output               // 29
                                                                       // 30
  // estimation                                                       // 31
  psi=[yt(t:-1:t-ord); 1];                  // reg. vector            // 32
  V=V+psi*psi';                             // updt of inf. matrix    // 33
  Vy=V(1,1); Vyp=V(2:$,1); Vp=V(2:$,2:$);   // division of inf. matrix // 34
  thE(:,1)=inv(Vp+1e-8*eye(Vp))*Vyp;        // est. of reg. coeff     // 35
```

```
  r(t)=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/t; // est. of noise variance  // 36
  Et(:,t)=thE;                               // stor est. parameters    // 37
end                                                                    // 38
                                                                       // 39
// Results                                                             // 40
// evolution of parametrs                                              // 41
set(scf(1),'position',[100 100 1200 400]);                             // 42
subplot(121),plot(Et')                                                 // 43
set(gca(),"data_bounds",[0 nd+1 -1 5])                                 // 44
title('Evolution of estimated parameters')                             // 45
legend('b0E','a1E','b1E','a2E','b2E','kE');                            // 46
// comparison of yt and yp                                             // 47
subplot(122)                                                           // 48
s=1:length(yt);                                                        // 49
plot(s,yt(s),'.:',s,yp(s),'rx')                                        // 50
set(gca(),"data_bounds",[1 nd -4 max(yt)+5])                           // 51
legend('output','prediction');                                         // 52
title([string(np),'- step ahead prediction'])                          // 53
// Relative Prediction Error                                           // 54
RPE=variance(yt(s)-yp(s))/variance(yt(s))                              // 55
```

Description of the program

- Row 2 defines number of steps of the prediction
- Row 3 sets the order of the regression model used
- Rows 5–7 define the interval of data used
- Row 10 loads the data from disk
- Rows 13–15 specify the prior information and initial data for the model
- Rows 18–39 perform the time loop.
  As we predict when y(t) is not measured, yet, the sequence of the tasks is: prediction (estimate of y(t)), simulation (measuring of y(t), estimation (with newly measured data)).
  - 20: construction of regression vector
  - 21: first step of prediction (based on measured data)
  - 22–26: loop for internal predictions (all unknown values of y are substituted by their estimates from the previous steps)
    * 23: construction of regression vector (predictions of y from previous steps are included).
    * Note: For the shift of the regression vector see the head of the program.
    * 24: generation of the new prediction
  - 26: the final prediction at the time t+np
  - 29: data measurement (from the loaded dataset)

- Row 32 extended regression vector for estimation
- Row 31 update of information matrix (if we do not estimate the noise covariance, the counter is not need)
- Rows 33–34 division of information matrix
- Row 35 point estimate of regression coefficients