```
// T24preREg2.sce
// N-STEP PREDICTION WITH CONTINUOUS MODEL (WITH ESTIMATION)
// Experiments
// Change: - np = number of steps of prediction
//         - r  = noise variance (effect on estimation)
//         - th = model parametrs
//         - u  = input signal (effect on estimation)
//  Try to define different model odrer for simulation and estimation
// ---------------------------------------------------------------
exec("ScIntro.sce",-1), mode(0)

nd=100;                                 // number of data              // 1
// PARAMETERS                                                         // 2
np=5;                                   // length of prediction (np>=1) // 3
nz=3;                                   // starting time (ord+1)       // 4
//  b0 a1 b1  a2 b2 k                                                  // 5
thS=[1 .4 -.3 -.5 .1 1]';               // regression coefficients     // 6
rS=.2;                                  // noise variance              // 7
u=signal(nd,1);                         // input                       // 8
y(1)=1; y(2)=-1;                        // prior data                  // 9
thE=rand(6,1,'n');                      // prior parametrs             // 10
nu=zeros(4,2);                                                         // 11
V=1e-8*eye(7,7);                        // initial information matrix  // 12
```

```
                                                                    // 13
// TIME LOOP                                                        // 14
for t=nz:(nd-np)                          // time loop (on-line tasks)   // 15
  // prediction                                                     // 16
  ps=[u(t) y(t-1) u(t-1) y(t-2) u(t-2) 1]'; // regression vector     // 17
  yy=ps'*thE;                             // first prediction at t+1     // 18
  for j=1:np                              // predictions for t+2,..,t+np // 19
    ps=[u(t+j); yy; ps(1:$-3); 1];       // reg.vecs with pred. outputs // 20
    yy=ps'*thE;                           // new prediction (partial)    // 21
  end                                                              // 22
  yp(t+np)=yy;                            // final pred. for time t+np   // 23
                                                                   // 24
  // simulation                                                    // 25
  ps=[u(t) y(t-1) u(t-1) y(t-2) u(t-2) 1]'; // regression vector     // 26
  y(t)=ps'*thS+sqrt(rS)*rand(1,1,'n');      // output generation     // 27
                                                                   // 28
  // estimation                                                    // 29
  Ps=[y(t) u(t) y(t-1) u(t-1) y(t-2) u(t-2) 1]';  // ext.reg.vect.  // 30
  V=V+Ps*Ps';                             // update of statistics      // 31
  Vp=V(2:$,2:$);                                                   // 32
  Vyp=V(2:$,1);                                                    // 33
  thE=inv(Vp+1e-8*eye(Vp))*Vyp;           // point estimates           // 34
  Et(:,t)=thE(:,1);                       // stor est. parameters      // 35
```

```
end                                                                    // 36
                                                                       // 37
// RESULTS                                                             // 38
disp(' Simulated parameters')                                          // 39
disp(thS)                                                              // 40
disp(' Estimated parameters')                                          // 41
disp(thE)                                                              // 42
                                                                       // 43
set(scf(1),'position',[500 100 1200 400]);                             // 44
subplot(121),plot(Et')                                                 // 45
set(gca(),"data_bounds",[0 nd+1 -1 2])                                 // 46
title('Evolution of estimated parameters')                             // 47
subplot(122)                                                           // 48
s=(np+3):(nd-np);                                                      // 49
plot(s,y(s),'.:',s,yp(s),'rx')                                         // 50
set(gca(),"data_bounds",[1 nd -3 5])                                   // 51
legend('output','prediction');                                         // 52
title(string(np)+'-steps ahead prediction')                            // 53
```

Description of the program

- Row 3 defines number of steps of the prediction
- Row 4 sets the starting time (the regression model needs some delayed values - as we cannot use zero or negative

indexes, we must start indexing at 1 and the first time index (denoting the time t=1) will be higher. Here, for second order model we need two initial y, thus the starting time is 3.

- Rows 5–7 prepare values of parameters
- Row 7 defines the input variable
- Rows 9–12 set initial y and statistics
- Rows 15–36 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)).
  - 17: construction of regression vector
  - 18: first step of prediction (based on measured data)
  - 19–22: loop for internal predictions (all unknown values of y are substituted by their estimates from the previous steps)
    - 20: 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.
    - 21: generation of the new prediction
  - 23: the final prediction at the time t+np
  - 22: regression vector for simulation
  - 23: simulation of new output y(t)
  - Row 30 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 32–33 division of information matrix
- Row 34 point estimate of regression coefficients