

```

// Kalmanův filtr - nelineární
//   x1(t) = x1(t-1)*x2(t-1)+u(t)
//   x2(t) = 0.5*x1(t-1)+0.8*x2(t-1)-u(t)
//   y(t) = x1(t-1)
// -----
exec("ScIntro.sce",-1),
getd(), mode(0)

nd=60; // number of steps // 1
// PARAMETERS // 2
xs=[2 -1]'; // initial state // 3
swS=.01* [.1 0; -.05 .1]; // state covariance // 4
svS=.0001; // output variance // 5
// 6
// SIMULATION // 7
for t=2:nd // 8
    u=.2*rand(1,1,'n'); // input // 9
    w=rand(2,1,'n'); // state noise // 10
    x(1)=xs(1)*xs(2)+.1*xs(2)+u+swS(1,:)*w; // state // 11
    x(2)=.5*xs(1)+.8*xs(2)-u+swS(2,:)*w; // generation // 12
    y=x(1)+svS*rand(1,1,'n'); // output generation // 13
    xt(:,t)=x; yt(t)=y; ut(t)=u; // remember for plot // 14
    xs=x; // for the next period // 15

```

```

end // 16
// 17
// ESTIMATION // 18
// initialization // 19
rxE=1e5*eye(2,2); // init. state covariance // 20
rwE=sWS'*sWS; // covariance of state // 21
rvE=svS**2; // covariance of output // 22
z=xs; // init. state estimate // 23
// time loop for estimation // 24
for t=2:nd // 25
    M=[z(2) z(1); .5 .8]; // lin.mod. parametrs // 26
    N=[1 -1]'; // 27
    F=[z(1)*z(2)+ut(t); .5*z(1)+.8*z(2)-ut(t)]+M*z // 28
    A=[0 1]; // 29
    B=0; // 30
    G=0; // 31
    [z,rxE,yp]=Kalman(z,yt(t),ut(t),M,N,F,A,B,G,rwE,rvE,rxE); // Kalman // 32
    zt(:,t)=z; // remember old state // 33
    yyp(t)=yp; // remember predicted y // 34
end // 35
// 36
// RESULTS // 37
s=1:nd; // 38

```

```
plot(s,xt(1,s),s,z(1,s),'.') // 39  
legend('state','estimate'); // 40
```

Description of the program