# Odhad stavu s nelineárním stavovým modelem

- stavový model se známými parametry
- nelineární model
- simulovaná data

V programu se provádí simulace s nelineárním stavovým modelem.

Pro odhad stavu je tento model linearizován pomocí prvých dvou členů Taylorova rozvoje. Linearizovaný model je využit pro odhad stavu z generovaného vstupu a výstupu. Odhad se provádí pomocí Kalmanova filtru. Kalmanův filtr je realizován procedurou

[x, xf, rx, yp]=Kalman(x, yt, ut, M, N, F, A, B, G, rw, rv, rx)

kde x je přepočítávaný stav, xf je výsledný odhad stavu v daném čase, rx je přepočítávaná kovariance stavu, yp je predikce výstupu, yt a ut jsou data, M, N, F, A, B, G jsou parametry stavového modelu, rw a rv jsou kovariance stavu a šumu ve stavovém modelu.

Použitý model má tvar

$$\begin{aligned} x_{1;t+1} &= x_{1;t}x_{2;t} - u_t + w_{1;t} \\ x_{1;t+1} &= 0.5x_{1;t} + 0.8x_{2;t} + u_t + w_{2;t} \\ y_t &= x_{1;t} \end{aligned}$$

kde $x_t$ je v programu označen jako xf

**Předpoklady:** Známe parametry modelu, linearizace modelu pomocí Taylorova rozvoje

**Sci značení:** $x_{t-1}/x_t/x_{t+1}$ - x, $x_t$ - xf, (ostatní viz Kalmanův filtr nahoře)

**Úloha:** Odhad neznámé veličiny, filtrace zašuměného signálu.

### Poznámka

*Pro správný start odhadování je důležité správné nastavení kovariančních matic. Matice rw se nastavuje většinou jako diagonální s velkými čísly na diagonále ($10^3 - 10^5$). Kovarianční matice rw a rv by měly odpovídat kovariančním maticím šumů $w_t$ a $v_t$ z modelu. POZOR: nejsou to kovarianční matice stavu a výstupu ale jejich poruch. Na správném nastavení těchto matic velmi záleží celý odhad.*

### Doporučené experimenty

1. Měňte rozptyly šumů *rw* a *rv* a sledujte jejich efekt na kvalitu odhadu.
2. Zkuste změnit simulovanou soustavu. Ale pozor! Pro novou soustavu musíte přepočítat matice $M$, $N$, $A$, $B$, $F$ a $G$ které plynou z linearizace nelineárního modelu.

**Program**

```
// State estimation (nonlinear model)
[u,t,n]=file();                          // find working directory
chdir(dirname(n(1)));                    // set working directory
clear("u","t","n")                       // clear auxiliary data
exec("ScIntro.sce",-1),mode(0)           // intro to sesion

nd=200;                                   // length of data

ut=.1*(1-rand(1,nd,'norm'));             // control
x1=0; x2=-1;                             // initial state

xs=zeros(2,nd); xs(:,1)=[x1;x2];
yt=zeros(1,nd);

// SIMULATION
for t=2:nd
  x1=x1*x2+.1*x2-ut(t)+.01*rand(1,1,'norm');  // state
  x2=.5*x1+.8*x2+ut(t)+.01*rand(1,1,'norm');  //     equations
  yt(t)=x2;                              // output equation
  xs(:,t)=[x1;x2];                     // store the actual sate variable
end

xx=[0;0];                                // initial state estimate
rw=.1*eye(2,2); rv=.1; rx=1e3*eye(2,2); // model and initial covariances
xt=zeros(2,nd); rr=zeros(2,nd); rr(:,1)=diag(rx);
// ESTIMATION
for t=2:nd
  // parameters of the linearized model
  M=[xx(2) xx(1)+.1; .5 .8];
  N=[-1;1];
  F=[-.1*xx(1)+(.1-xx(2))*xx(2);0];
  A=[0 1];
  B=0;
  G=0;
  // Kalman filter
  [xx,rx,yp]=Kalman(xx,yt(t),ut(t),M,N,F,A,B,G,rw,rv,rx);
  xt(:,t)=xx;                            // stor actual state estimate
  rr(:,t)=diag(rx);                      // stor variance of noise estimate
end

// RESULTS
// figure 1
plot(xs')
plot(xt',':')
legend('state 1','state 2','estimate 1','estimate 2',4);
title('State and its estimate','fontsize',5,'FontName','Times')
xlabel('time','fontsize',4,'FontName','Times')
```

```
ylabel('values','fontsize',4,'FontName','Times')
set(gcf(),'position',[300 100 500 400])
```