Simulace s regresním modelem

Úloha: Simulace dat pro další použití.

- jednorozměrný vstup i výstup
- obecný řád modelu

Simulují se data z normálního, jednorozměrného regresního modelu

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_n y_{t-n} + b_0 u_t + b_1 u_{t-1} + \cdots + b_n u_{t-n} + k + e_t$$

kde y, u, e jsou výstup, vstup a porucha, a, b, k jsou parametry.

Předpoklady: $e \sim N(0, r)$, r konstantní; vstup je generován předem.

Poznámka

Např. pro testování nových algoritmů je simulace dat velice důležitá. Data je totiž možno simulovat podle vlastních představ tak, aby byly otestovány všechny možné varianty použití nového algoritmu.

Závěrečné testování je ale dobré provést na reálných datech, aby algoritmus byl co nejlépe připraven k použití v reálném prostředí.

Značení

- y - yt,
- u - ut,
- a, b, k - a, b, k, (použití viz níže),
- r - cv (použití viz níže).

Volitelné parametry

- nd - počet simulovaných dat
- Sim.Cy.ord - řád modelu
- I_typU - volba řízení
- Sim.Cy.tha1
  Sim.Cy.thb1
  Sim.Cy.thk
  Sim.Cy.cv - parametry modelu
  (značení: Sim - simulace, Cy - komponenta pro spojitý výstup)

Doporučené experimenty

1. Měňte parametry soustavy tak, aby byla (i) pomalá, (2) rychlá.

   - Rychlostí se myslí délka odezvy na jednotkový vstup.
   - Tato rychlost je dána velikostí vlastních čísel charakteristické rovnice.
   - U prvního řádu je to velmi jednoduché: pro tha blízké nule dostaneme soustavu rychlou, pro tha blízké jedné je soustava pomalá.
   - Abychom rychlost soustavy mohli pozorovat, volíme vstup roven jedné a hodně malý rozptyl šumu.

2. Zkuste měnit řád modelu. Pozor: předvolby parametrů jsou nastaveny jen do řádu 5. Potom je potřeba další koeficienty dodat. Nicméně, viditelné rozdíly jsou patrné tak do řádu 3.

3. Zajímavé je poexperimentovat s řádem 2. Nastavit soustavu s reálnými kořeny, jedním dvojnásobným kořenem a s komplexními kořeny. Vlastnosti jsou dány řešením charakteristické rovnice.

Program

```
// Simulation of scalar regression model of order n
//
[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
deff('ps=genpsi(t,n,y,u)','ps=[y(t-(1:n)),u(t-(0:n)) 1]','c')

nd=100;                                  // number of data
I_typU=1;                                // type of input

// model
Sim.Cy.ord=2;                            // model order
Sim.Cy.tha=[.6 -.2 .1 -.5 .2];           // parameters at y
Sim.Cy.thb=[1 .5 -.3 -.1 .1 -.3];        // parameters at u
Sim.Cy.thk=-1;                           // model constant
Sim.Cy.cv=.01;                            // variance of noise

// model of input signal
select I_typU                            // choice of input
  case 1, ut=rand(1,nd,'u');             // random, uniform
  case 2, ut=ones(1,nd);                  // ones
  case 3                               // two steps
    ut=[ones(1,fix(nd/2)), -ones(1,fix(nd/2)+1)];
  case 4                                 // random jumps
    ut(1)=1; j=1;
    for i=2:nd
      if rand(1,1,'u')>.85, j=rand(1,1,'n'); end
      ut=[ut j];
```

```
    end
  case 5, ut=abs(sin(10*(1:nd)/nd));    // sinus
end
yt=zeros(1,nd);
ord=Sim.Cy.ord;
cv=Sim.Cy.cv;
th=[Sim.Cy.tha(1:ord) Sim.Cy.thb(1:(ord+1)) Sim.Cy.thk];
                                        // regression coefficients
Sim.Cy.th=th;

// time loop
for t=(ord+1):nd                        // time loop
  ps=genpsi(t,ord,yt,ut);               // regression vector
  yt(t)=ps*th'+cv*rand(1,1,'n');        // model simulation
end
Sim.Cy.yt=yt;
Sim.Cy.ut=ut;

// results
s=1:nd;
plot(s,yt(s),s,ut(s))
legend('output','input');
title('Simulation with regression model')
set(gca(),'data_bounds',[1 nd min([yt,ut])-.1 max([yt,ut])+.1])

save _data/dataT11.dat Sim
```