Řízení s kategorickým modelem

- model řízené mince s pamětí
- řízení na jednom intervalu

Použitý kategorický model

$[u_t, y_{t-1}]$	$y_t = 1$	$y_t = 2$
1, 1	$\Theta_{1\|11}$	$\Theta_{2\|11}$
1, 2	$\Theta_{1\|12}$	$\Theta_{2\|12}$
2, 1	$\Theta_{1\|21}$	$\Theta_{2\|21}$
2, 2	$\Theta_{1\|22}$	$\Theta_{2\|22}$

Předpoklady: Pevné a známé parametry modelu, řízení na jednom intervalu.

Sci značení: om penalizace, f $= E[fp\|d(t)]$ - expectation over $y(t+1)$ fs $= \min f \rightarrow u(t+1)$ - optimal u

Úloha: Řízení s kategorickým modelem se známými parametry.

Poznámka

Realizuje se optimální s obecnou penalizací jednotlivých konfigurací veličin vystupujících v modelu (rozšířený regresní vektor).

Doporučené experimenty

1. Penalizace s v případě řízení s diskrétním modelem volí ve tvaru matice (stejných rozměrů jako u modelu) a penalizuje se každý stav (tj. kombinace veličin y_t, u_t, y_{t-1}) zvlášť. Určete si svou vlastní preferenci pro řízení, realizujte ji penalizační tabulkou *Con.Cz.om* a sledujte, jak je ve výsledku realizována.
2. Zkuste měnit simulovanou soustavu tak, aby v ní bylo více/méně neurčitosti. Sledujte kvalitu řízení.

Program

```
// Dynamic programming with discrete model (single horizon)
//  fp = omega + fs  (penalty function, old criterion)
//  f  = E[fp|d(t)] - expectation over zt(t+1)
//  fs = min f  =>  ut(t+1) - optimal ut
[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

// VARIABLES TO BE SET
```

```
Con.Cz.nh=30;                          // length of control interval
I_detMod=0;                           // 1 for detrministic model: round(th)
y0=1;                           // initial condition for output

//        zt 1 2          ut y1  =  criterion
Con.Cz.om=[0 5        // 1 1
           0 5        // 1 2
           5 0        // 2 1
           5 0];      // 2 2

//       zt  1  2       ut y1  =  system model
Sim.Cz.th=[.2 .8     // 1 1
           .9 .1     // 1 2
           .1 .9     // 2 1
           .8 .2];   // 2 2
if I_detMod==1
  th=round(th);               // deterministický model
end

// computed variables and initializations
fs=zeros(1,2);
om=Con.Cz.om;
nh=Con.Cz.nh;
th=Sim.Cz.th;

// CONTROL LAW COMPUTATION
for t=nh:-1:1
  fp=om+ones(4,1)*fs;         // penalty + reminder from last step
  // expectation
  f=sum((fp.*th),'c');          // expectation over zt
  // minimization
  if f(1)<f(3),               // for zt(t-1)=1
    us(t,1)=1; fs(1)=f(1);    // optimal control, minimum of criterion
  else
    us(t,1)=2; fs(1)=f(3);    // optimal control, minimum of criterion
  end
  if f(2)<f(4),               // for zt(t-1)=2
    us(t,2)=1; fs(2)=f(2);    // optimal control, minimum of criterion
  else
    us(t,2)=2; fs(2)=f(4);    // optimal control, minimum of criterion
  end
end

// CONTROL APPLICATION
zt(1)=y0;
for t=1:nh
  ut(t+1)=us(t,zt(t));          // optimal control
  i=2*(ut(t+1)-1)+zt(t);        // row in the model table
  zt(t+1)=(rand(1,1,'unif')>th(i,1))+1;    // simulation
```

```
end
Con.Cz.zt=zt;
Con.Cz.ut=ut;

// RESULTS
plot(1:nh+1,zt,'ro')
plot(1:nh+1,ut,'g+','markersize',14)
set(gcf(),'position',[200 40 1000 600])
set(gca(),"data_bounds",[.8 nh+.2, .8 2.2])
title('Optimal control with discrete model')
legend('output','input');
```