```
%-----------------------------------------------------------------
%       Rotary Motion Inverted Pendulum System  for Quanser JP
%
%       This program is nonlinear system
%
%       Motor  input -24V <------> +24V
%        振り上げのみのプログラム+LQGコントローラ
%
%       Notice!
%       call swingup12_eq12.m
%       call swingup11_LQG(simulink program)
%
%           2006.7.26                          produced by kizawa
%-------------------------------------------------------------------------------

clc
clear all
format long e

global MD

MD=zeros(2,2);

%----setup---%
sampling=0.001;%sampling time of simulation
kaisu=1000;
timespan=sampling*kaisu;% simulation time
%--- initial state ----
th0=0.00; %angle of arm(rad)
th1= -1*pi; %angle of pendulum(rad)          -3.14:真下
d_th0=0.0; %angle velocity of arm(rad/s)
d_th1=0.0; %angle velocity of pendulum(rad/s)

x0=[th0 th1 d_th0 d_th1]; %initial condition
%manipulated variable
Vin=0.0; %[Volt]
u2=0.0;%%%%%%%%


%---mechanical parameter---
mp = 0.067;     %Mass of Pendulum[Kg]半分0.066
ma = 0.057;      %Mass of Arm[Kg]
r = 0.137;      %length of Arm[m]
ra= 0.0685;      %length to the center of gravity of arm[m]
lp= 0.155;      %length to the center of gravity of pendulum[m]0.167
Lp=0.317;     %length of pendulum
Ia=2.67e-4;      %arm inertia moment of the circumference of the center of gravity1.2e-3
Ip=0.971e-3;      %pendulum inertia moment of the circumference of the center of gravity
Jm=2.19e-6;    %inertia moment of moter[kgm^2]
Ra = 5.7;      %armature resistance[ORM]
Km = 3.038e-2;    %Motor torque constant0.023[Nm/A]
Ke =3.06e-2;  %[Vrad/s]
ca=2.919e-3;   %[Nms/rad]armエンコーダ0.2016摩擦を小さく0.04
cp=1.671e-4;  %[Nms]pendulumエンコーダ 1.899e-4
n = 29.47;     %gear ratio 79.24
g = 9.8;       %[m/s^2] acceleration of gravity
```

```
Jb=1.2e-4;%[kgm^2]アームと取り付け部を含むInertia;1.15e-4
Kti=n*Km/Ra;
cti=ca + n^2*Ke*Km/Ra;


%-----------------  Non linear system--------------------------------------------
%---definition---
%アーム角度  振子角度  アーム角速度  振子角速度  アーム角加速度  振子角加速度   電圧
%---y(1,1)   y(2,1)     y(3,1)       y(4,1)                                      u
%                      dy(1,1)      dy(2,1)     dy(3,1)         dy(4,1)
%-------------------------------------------------------------------------------------

%---Y(1,1)      Y(2,1)     Y(3,1)       Y(4,1)       DDY(1,2)          DDY(2,1)
%アーム角度    振子角度  アーム角速度  振子角速度      アーム角加速度  振子角加速度
%-------------------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%-------------Liner System
------------------------------------------------------------------------%
E=[Jb + mp*r^2  + Jm*n^2       mp*r*lp
   mp*r*lp                              mp*lp^2 + lp    ];%l1 + m1*l1^2   pendulum inertia around
rotary shaft

A1=[-ca - n^2*Ke*Km/Ra      0
     0                     -cp];

A2=[0      0
    0      mp*g*lp];

B1=[n*Km/Ra  0]'; %間違い

A=[zeros(2,2)     eye(2,2)
   inv(E)*A2   inv(E)*A1];

B=[zeros(2,1)
   inv(E)*B1];


C=[ 1  0   0  0
    0  1   0  0];


D=zeros(2,1);
%---------------------------- LQG/LTR
Controller-----------------------------------------------------------%
%----LQ Controller---%
Q = [1000       0       0      0           %1
      0        10.0       0      0          %10
      0        0          10     0           %1
      0        0          0      3];        %3

R = [0.4];   %0.0025  0.1

qq=1e3; %1e3
```

```
PP=care(A,B,Q ,R);      %Ricutti equation
disp('FeedbackGain')
Fgain=inv(R)*B'*PP      %feedback gain  [V/rad]
disp('Eigen value A-BF')
eigen_Af=eig(A-B*Fgain)

%K=K/180*3.14;           %feedback gain  [V/degree]
SS=care(A',C', qq * B * B' );  %Ricutti equation
disp('ObseberGain')
Kalgain=SS * C'              %kalman felter gain
disp('Eigen value A-Kalgain * C')
eigen_Akal=eig(A-Kalgain*C)

Ak=A - Kalgain * C - B * Fgain;
Bk=Kalgain;
Ck= -Fgain;
Dk=[0 0];
disp('Controller  Eigen value     A - Kalgain * C -B * Fgain ')
eigen_Ak=eig(Ak)


disp('コントローラの固有値は')
eig(Ak)

%--- 閉ループ系 -----%
disp('閉ループ系の固有値は')
AG=[A      B*Ck
    Bk*C   Ak  ];
eig(AG)
%-----------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----bounary condition---
ipu=1.0e-1; %0.1
delta=0.02;  %0.04 0.03
Vmax=24;%24[V]10

%-------------------------
disp('It is under calculation now!!')
ni=1;
tend=5000;  %7000
while(ni <= tend)
    %x0(1,3)

          %%%%%%%% Jump to  LQG/LTR Control for Liner System  %%%%%%%%%%%%%%
          %この条件も-360に収束するときは変更が必要,レギュレータも-360に収束することを検討する必↙
要がある
          if (   ( x0(1,2) > -pi/6 ) && ( x0(1,2) <  + pi/6 )   )   %-pi/6 この条件では厳しい
              break;
          end
          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


          if     ( -ipu <= x0(1,3) ) && ( x0(1,3)<=ipu )
                   if x0(1,3) >= 0
```

```
                Vin= Vmax * (+1);
          elseif x0(1,3) < 0
                Vin= Vmax * (-1);
          end

      else

          if x0(1,3) >= 0.0

               u2=( cp*x0(1,4)^2/x0(1,3) )  + delta;
               Vin=( u2 + cti*x0(1,3) )/Kti ;
               if Vin >=Vmax
                   Vin=+Vmax;
               end
          elseif x0(1,3) < 0.0

               u2=( cp*x0(1,4)^2/x0(1,3) )  - delta;
               Vin=( u2 + cti*x0(1,3) )/Kti ;
               if Vin <= -Vmax
                   Vin= -Vmax;
               end
          end
      end




     %%%%%%%%%%%%%%%%%%%アームが回り過ぎないように%%%%%%%%%%%%%%%%%%%5
     if     ( x0(1,1) > pi*0.5 )                 %スイングを規制すればエネルギが一端下がる0.7
             Vin=-Vmax;  %-1.3

     end
     if     ( x0(1,1) < -pi*0.5 )
             Vin=+Vmax;   %+1.3

     end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      [Tn,Yn]=ode45(@swingup_12_eq12,[0 sampling],x0,[],Vin,ca,cp,mp,r,lp,Ip,Jb,Ra,Km,Ke,Jm, ↙
n,g);


      if ni==1
         YY(1,:)=Yn(1,1:4);
         TT(1,:)=Tn(1,1);
         VIN(1,:)=Vin;
         U2(1,:)=u2;
         %---motion enegy---%
         ENE(1,1)=0.5*[Yn(1,3)  Yn(1,4) ] * MD * [Yn(1,3)  Yn(1,4)]' + mp*g*lp*cos( Yn(1,2) ↙
);
         %-- Dot  motion enegy ---%
         DENE(1,1)=u2*Yn(1,3) - cp*( Yn(1,4) )^2;
      else
        YY(ni,:)=Yn(size(Yn,1),1:4);
        TT(ni,1)=sampling*ni;
        VIN(ni,1)=Vin;
```

```
            U2(ni,:)=u2;
            %---motion enegy---%
            ENE(ni,1)=0.5*[Yn(1,3)  Yn(1,4)] * MD * [Yn(1,3)  Yn(1,4)]' + mp*g*lp*cos( Yn(1,2) ↙
);
            %-- Dot  motion enegy ---%
             DENE(ni,1)=u2*Yn(1,3) - cp*( Yn(1,4) )^2;
          end
          %---- Up data------
          x0=Yn(size(Yn,1),1:4);
          ni=ni+1;
end


ni

figure(1)
subplot(2,2,1)
plot(TT,180/pi*YY(:,1));
axis([0  sampling*ni -360  +360])
xlabel('Time[s]');
ylabel('Angle of Arm[degree]')

subplot(2,2,2)
line_bottom=-180*ones(1,size(YY,1));
line_top0=zeros(1,size(YY,1));
line_top1=-360*ones(1,size(YY,1));
line_dome1=30*ones(1,size(YY,1));
line_dome2=-30*ones(1,size(YY,1));
line_dome3=-330*ones(1,size(YY,1));
line_dome4=-390*ones(1,size(YY,1));

plot(TT,180/pi*YY(:,2),'-k',TT,line_bottom,'-.r',TT,line_top0,'-r',TT,line_top1,'-r',.......
     TT,line_dome1,'-b',TT,line_dome2,'-b',TT,line_dome3,'-b',TT,line_dome4,'-b');
axis([0  sampling*ni -450 +90])
xlabel('Time[s]');
ylabel('Angle of pendulum[degree]')
title('¥theta=0(deg),¥theta=-360(deg) ===>TOP')

subplot(2,2,3)
plot(TT,VIN);
axis([0  sampling*ni -25 +25])
xlabel('Time[s]');
ylabel('INPUT  Moter Voltage [V]')

subplot(2,2,4)
plot(TT,U2);
%axis([0  sampling*ni -1 +2.0])
xlabel('Time[s]');
ylabel('Input u2 ')

figure(2)
subplot(2,2,1)
plot(TT,YY(:,3));
axis([0  sampling*ni -15  +15])
xlabel('Time[s]');
ylabel('Angular velocity[rad/s]')
```

```
title('Angular velocity  of Arm')

subplot(2,2,2)
plot(TT,YY(:,4));
axis([0  sampling*ni -15  +15])
xlabel('Time[s]');
ylabel('Angular velocity[rad/s]')
title('Angular velocity  of Pendulum')

subplot(2,2,3)
zeroline=zeros(1,size(YY,1));
plot(TT, DENE, TT,zeroline,'-r');
axis([0  sampling*ni -1  +6])
xlabel('Time[s]');
ylabel('Dot Energy')
title('Dot Energy ')

subplot(2,2,4)
plot(TT,ENE);
axis([0  sampling*ni -1 +3.0])
xlabel('Time[s]');
ylabel('Energy [J]')

%%%%%%%%  LQG/LTR Control for Liner System  %%%%%%%%%%%%%%
%==== Call Simulink=======
%x0(1,3)=0;
%x0(1,4)=0;          %これがゼロであれば制御が容易
XX0=x0;              %initial value for simulink  速度を持ってこの領域に入ることに注意
modelname='swingup12_LQG';               %change
FixedStep=sampling;
OPTIONS=SIMSET('FixedStep',sampling);             %sampling set for simulink
timespan=[ni*sampling    tend*sampling]          %timespan=[tStart  tFinal]
tt=sim(modelname,timespan,OPTIONS);
%=============================================================
%---motion enegy---
%ENEpuls(1,1)=0.5*[Yn(1,3)  Yn(1,4)] * MD * [Yn(1,3)  Yn(1,4)]' + mp*g*lp*cos( Yn(1,2) );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

YYY1=vertcat( YY(:,1),TH0);                       %アーム角度  ベクトル垂直連結
TTT=vertcat( TT,tt);
YYY2=vertcat( YY(:,2),TH1);                       %振子角度   ベクトル垂直連結

YYY3=vertcat( YY(:,3),THV0);                       %アーム角速度  ベクトル垂直連結
YYY4=vertcat( YY(:,4),THV1);                       %振子角速度   ベクトル垂直連結

%---transient response for nonliner system and  liner system---
figure(3)
subplot(2,2,1)
plot(tt, [180/pi*TH0  180/pi*TH1])
ax1=findobj(gca,'Type','Axes','Visible','on');
al1=findobj(gca,'Type','Line','Visible','on');
set(al1(1),'LineStyle','-')
set(al1(2),'LineStyle',':')
axis([ni*sampling tend*sampling -100  100])
set(ax1,'fontsize',12)                            %legend
%title('  Transient Response with initial condition [th0 th1]')
```

```
xlabel('TIME(s)','FontSize',12)
ylabel('Angle(degree)','FontSize',12)
legend('Arm','Pendulum',1)
grid on

figure(3)
subplot(2,2,2)
plot(TTT,YYY3(:,1));
axis([ni*sampling  sampling*tend -15  +15])
xlabel('Time[s]');
ylabel('Velocity of Arm[rad/s]')
grid on

figure(3)
subplot(2,2,3)
plot(TTT,YYY4(:,1));
axis([ni*sampling  sampling*tend -15  +15])
xlabel('Time[s]');
ylabel('Velocity of Pendulum[rad/s]')
grid on

figure(4)
subplot(2,2,1)
plot(TTT,180/pi*YYY1(:,1));
axis([0  sampling*tend -200  +200])
xlabel('Time[s]');
ylabel('Angle of Arm[degree]')
grid on

figure(4)
subplot(2,2,2)
plot(TTT,180/pi*YYY2(:,1));
axis([0  sampling*tend -360  +60])
xlabel('Time[s]');
ylabel('Angle of Pendulum[degree]')
grid on
%----animation---%
figure(5)
xp=Lp*sin(YYY2);%pendulum  xp=Lp*sin(YY(:,2))
yp= Lp*cos(YYY2);           %Lp*cos(YY(:,2))

xa=r*sin(YYY1);%arm

hp=plot([0;0],[xp(1),yp(1)],'linewidth',2,'Marker','o')
%ha=plot([0;0.5],[xa(1),ya(1)],'linewidth',2,'Marker','o')
axis([-Lp-0.2 Lp+0.2 -Lp-0.2 Lp+0.2]);
for i=1 : size(TTT,1)
    %set(hp,'xdata',[0,xp(i)],'ydata',[0,yp(i)],'Erasemode','xor');
    set(hp,'xdata',[xa(i), xa(i)+xp(i)],'ydata',[0,yp(i)],'Erasemode','xor');
   % set(hp,'xdata',[0,xa(i)],'ydata',[0,ya(i)],'Erasemode','xor');
    drawnow
    pause(0.001);
    grid on
end

%----Animation avi---%
```

```
figure(6)
xp=Lp*sin(YYY2);%pendulum  xp=Lp*sin(YY(:,2))
yp= Lp*cos(YYY2);           %Lp*cos(YY(:,2))

xa=r*sin(YYY1);%arm


 h1=plot([0;0],[xp(1),yp(1)],'linewidth',2,'Marker','o');
 axis([-Lp-0.2 , Lp+0.2 , -Lp-0.2 , Lp+0.2]);
 %
 %

 aviobj=avifile('example2.avi');  %change
 for i=1 : size(TTT,1) %TTT
     set(h1,'xdata',[xa(i), xa(i)+xp(i)],'ydata',[0,yp(i)],'Erasemode','xor');
     drawnow
     %pause(0.0001);
     grid on
      frame=getframe(gca);
      aviobj=addframe(aviobj,frame);
 end
 aviobj=close(aviobj);

% [Name,Path]=uiputfile({'*.avi','AviFiles(*.avi)'},'AVI-fil Save As');

%if isequal(Name,0)|isequal(Path,0)
%else
%    filename=fullfile(Path,Name);
%    movie2avi(aviobj,filename,'compression','Cinepak');
%end
```