

10.6 MATLAB – Teljes fellendülési stratégia

C:\brachiation\teljes_strategia\teljes_strategia_main.m

1 of 2

```
clear all %comment this out in debug mode
close all
clc

global param
teljes_strategia_init

%necessary total energy
E_nec = total_energy(param.des_angle(1),param.des_angle(2),0,0);
E_40_angle=total_energy(40/180*pi,0,0,0);
param.W_nec=E_nec(3);
param.W_40_angle=E_40_angle(3);

%simulation
sol = rk4(@teljes_strategia,tf,dt,param.ICs);
t=sol.t;
n=length(t);
q1 = sol.y(1,:);
q2 = sol.y(2,:);
q1dot = sol.y(3,:);
q2dot = sol.y(4,:);

%animation window
r1 = L1*[sin(q1); -cos(q1)];
r2 = r1 + L2*[sin(q1+q2); -cos(q1+q2)];
r_des1 = L1*[sin(param.des_angle(1,1)); -cos(param.des_angle(1,1))];
r_des2 = r_des1 + L2*[sin(param.des_angle(1,1)+param.des_angle(1,2)); -cos(param.
des_angle(1,1)+param.des_angle(1,2))];
figure('name','Animation window')
plt1 = plot([0,r1(1,1)],[0,r1(2,1)],'-ko','linewidth',2); %first segment
hold on
plt2 = plot([r1(1,1),r2(1,1)],[r1(2,1),r2(2,1)],'-ro','linewidth',2); %second segment
plt3 = plot(r_des2(1,1),r_des2(2,1),'--rs','LineWidth',
2,'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',10);
axis equal
set(gca,'xlim',[-1.2,1.2],'ylim',[-1,1])

%animation controllers
sld = uicontrol('style','slider','position',[0 50 100 20],'sliderstep',[1/n
10/n],'value',1,'min',1,'max',n,'callback','animate');
tinfo = uicontrol('style','text','position',[270 50 100 20],'string','time: ');

%Total energy
m1=param.m1;
m2=param.m2;
L1=param.L1;
L2=param.L2;
xG1=param.xG1;
xG2=param.xG2;
I1=param.I1;
```

```
I2=param.I2;
g=param.g;
T(n)=0;
U(n)=0;
W(n)=0;

for i=1:n
    E2(i,:) = total_energy(q1(i),q2(i),q1dot(i),q2dot(i));
end

figure('name','TotalEnergy')
plot(E2(:,3))
```

```
global param flag

%mechanical parameters
m1 = 1.129;
m2 = 0.762;
I1 = 0.042857;
I2 = 0.0406244;
L1 = 0.28;
L2 = 0.28;
xG1 = 0.140526;
xG2 = 0.198104;
yG1 = 0;
yG2 = 0;
g = 9.81;

%energy based controller parameters
E=20;
P = 50;
D = 10;
strategy = 1;
des_angle(1)=-80*pi/180;
des_angle(2)=-40*pi/180;
q2des=des_angle(2);

%store values in a structure
param.m1 = m1;
param.m2 = m2;
param.I1 = I1;
param.I2 = I2;
param.L1 = L1;
param.L2 = L2;
param.xG1 = xG1;
param.xG2 = xG2;
param.yG1 = yG1;
param.yG2 = yG2;
param.g = g;
param.P = P;
param.D = D;
param.E=E;
param.q2des = q2des;
param.strategy = strategy;
param.des_angle=des_angle;

%simulation paramters

tf = 40; %final simulation time
dt = 0.01; %timestep (in case of explicit Euler method set it to dt = 0.0001)
q0 = [0/180*pi 0/180*pi]'; %initial conditions
qdot0 = [0 0]';
ICs = [q0; qdot0];
```

```
flag = 0; %error flag for integration
```

```
param.tf = tf;
```

```
param.dt = dt;
```

```
param.ICs = ICs;
```

```

function ydot = dpend_mincoord(t,y)

global param flag

m1 = param.m1;
m2 = param.m2;
I1 = param.I1;
I2 = param.I2;
xG1 = param.xG1;
xG2 = param.xG2;
L1 = param.L1;
g = param.g;
P = param.P;
D=param.D;
E=param.E;
q2d = param.q2des;

q1 = y(1);
q2 = y(2);
q1dot = y(3);
q2dot = y(4);

%parametric energy increase
if param.strategy == 1
    tau = -0.5519156816215348*sin(4.65*t) + ...
        0.042267469440000004*q1dot^2*sin((pi*sin(4.65*t))/5) + ...
        ((0.0406244 + 0.042267469440000004*cos((pi*sin(4.65*t))/5))*...
        (-3.6494559077400006*sin(q1) + ...
        (0.5519156816215348 + 0.5742381230589251*cos((pi*sin(4.65*t))/5))*...
        sin(4.65*t) + 0.24698413895007534*q1dot*cos(4.65*t)*...
        sin((pi*sin(4.65*t))/5) + ...
        0.36080445376262216*(cos(4.65*t))^2*sin((pi*sin(4.65*t))/5) - ...
        1.4808709828800002*sin(q1 + (pi*sin(4.65*t))/5))/...
        (0.14322220000000002 + 0.08453493888000002*cos((pi*sin(4.65*t))/5)) + ...
        1.48087098288*sin(q1 + (pi*sin(4.65*t))/5);
end

%energy based controller
if param.strategy ==2
    W1=total_energy(q1,q2,q1dot,q2dot);
    W12=W1(3);
    u =-P*(q2-q2d)-D*q2dot-min(20,max(-20,E*(W12-param.W_nec)*sign(q1dot)));
    tau=I2*u + L1*m2*q1dot^2*xG2*sin(q2) + g*m2*xG2*sin(q1 + q2) - ...
        ((I2 + L1*m2*xG2*cos(q2))*(I2*u + L1*m2*u*xG2*cos(q2) + g*L1*m2*sin(q1) + ...
        g*m1*xG1*sin(q1) - 2*L1*m2*q1dot*q2dot*xG2*sin(q2) - ...
        L1*m2*q2dot^2*xG2*sin(q2) + g*m2*xG2*sin(q1 + q2)))/...
        (I1 + I2 + L1^2*m2 + 2*L1*m2*xG2*cos(q2));
end

param.tau(param.n)=tau;

```

```
%equation of motion
m11=I1+I2+L1^2*m2+2*L1*m2*xG2*cos(q2);
m12=I2+L1*m2*xG2*cos(q2);
m22=I2;
c1=-L1*m2*xG2*sin(q2)*q2dot*(2*q1dot+q2dot);
c2=L1*m2*xG2*sin(q2)*q1dot^2;
Q1= -g*((L1*m2 + m1 * xG1)*sin(q1) + m2*xG2*sin(q1 + q2));
Q2=tau - g*m2*xG2*sin(q1 + q2);

M = [[m11,...
      m12];...
      [m12,...
      m22]];
c = [c1;...
      c2];
Q = [ Q1;...
      Q2];

qddot = M\ (Q - c);

qdot = y(3:4);

ydot = [qdot; qddot];

end
```

```
function sol = rk4(fhandle,tf,dt,ICs)

global flag param

%initial values
t = 0;
y = ICs;
n = 1;
N = length(ICs);
sol.t(n) = t;
sol.y(1:N,n) = y;

while t<tf && flag<1
    param.n = n;
    k1 = dt * feval(fhandle, t, y);
    k2 = dt * feval(fhandle, t + dt/2 , y + k1/2);
    k3 = dt * feval(fhandle, t + dt/2 , y + k2/2);
    k4 = dt * feval(fhandle, t + dt, y + k3);

    t = n*dt;
    y = y + (k1 + 2*k2 + 2*k3 + k4)/6;
    param.n=n;
    n = n+1;
    sol.t(n) = t;
    sol.y(1:N,n) = y;

    E1 = total_energy(y(1),y(2),y(3),y(4));
    W=E1(3);
    param.W1(n)=W;

    if W>param.W_40_angle && abs(y(2)) < 0.05 && param.strategy ==1
        param.strategy=2;
    end
    if abs(y(1)-param.des_angle(1,1))<0.05 && abs(y(2)-param.des_angle(1,2))<0.05
        t=tf;
    end
end
end
```

```
function [E] = W_nec( q1,q2,q1dot,q2dot )

global param

m1=param.m1;
m2=param.m2;
L1=param.L1;
L2=param.L2;
xG1=param.xG1;
xG2=param.xG2;
I1=param.I1;
I2=param.I2;
g=param.g;

rA1S1=[xG1*sin(q1) -xG1*cos(q1) 0]';
rA1A2=[L1*sin(q1) -L1*cos(q1) 0]';
rA2S2=[xG2*sin(q1+q2) -xG2*cos(q1+q2) 0]';
vs1=cross([0 0 q1dot]',rA1S1);
vs2=cross([0 0 q1dot]',rA1A2)+cross([0 0 q1dot+q2dot]',rA2S2);

T=1/2*(m1*vs1'*vs1+(I1-m1*xG1^2)*q1dot^2+m2*vs2'*vs2+(I2-m2*xG2^2)*(q1dot+q2dot)^2);
U=m1*xG1*g+m2*(L1+xG2)*g-m1*g*xG1*cos(q1)-m2*g*(L1*cos(q1)+xG2*cos(q1+q2));
W=U+T;
E=[T;U;W];
end
```

```
sld_value = get(sld,'value');

i = floor(sld_value);

set(plt1,'xdata',[0,r1(1,i)],'ydata',[0,r1(2,i)])
set(plt2,'xdata',[r1(1,i),r2(1,i)],'ydata',[r1(2,i),r2(2,i)])

set(tinfo,'string',['time: ' num2str(i*dt)])

drawnow
```