Homework 3 –solution

From the Hamiltonian equation, we can have that

Then .

So the value of u that minimize H is given by the signum function:

For this problem, the switch of u happens only once. Then the minimum time trajectory can be determined by different methods.

***Solution 1: (Phase plane approach)***

Plot the phase line for both u=+1 and u=-1, and then find the intersection point of the two trajectories. This intersection point can be found by different ways. For example, you can try to find it from the figure, or you can solve the equations and get the analytical or numerical solution.

The intersection point is

The minimum time required is



Appendix –MATLAB codes

function dx=fun\_pos(t,x)

dx=zeros(2,1);

dx(1)=-x(2);

dx(2)=1-0.05\*(x(2)^2);

function dx=fun\_neg(t,x)

dx=zeros(2,1);

dx(1)=-x(2);

dx(2)=-1-0.05\*(x(2)^2);

[t\_pos,x\_pos]=ode45(@fun\_pos,[0 30],[100 3]);

[t\_neg,x\_neg]=ode45(@fun\_neg,[0 -5],[0 0.05]);

***Solution 2: Non-linear programming method***

This problem can also be solved by non-linear programming method. You can use the “fmincon” function in MATLAB. First, give some guess of the value U and DT, then try to find the value of U and DT which can minimize the objective function.





% This code is written by Kody Powell.

global Z0;

x0=100;

v0=3;

Z0=[x0;v0];

U=[1;-1;1;-1];

DT=[10;5;6;8];

UDTguess=[U;DT];

LB=[-1;-1;-1;-1;0.0001;0.0001;0.0001;0.0001];

UB=[1;1;1;1;30;30;30;30];

[Uopt,J]=fmincon('boat\_obj',UDTguess,[],[],[],[],LB,UB);

U=Uopt(1:4);

DT=Uopt(5:8);

T=[DT(1);sum(DT(1:2));sum(DT(1:3));sum(DT)];

[t,Z]=ode45(@(t,Z) boat(t,Z,U(1)),[0,T(1)],Z0);

for i=2:4

[t1,Z1]=ode45(@(t,Z) boat(t,Z,U(i)),[T(i-1) T(i)],Z(end,:)');

t=[t;t1];

Z=[Z;Z1];

end;

figure(1);

subplot(3,1,1)

plot(t,Z(:,1))

ylabel('x (m)');

xlabel('time (s)');

axis([0 T(end) -5 105]);

subplot(3,1,2)

plot(t,Z(:,2))

ylabel('v (m/s)');

xlabel('time (s)');

axis([0 T(end) -0.1 5]);

subplot(3,1,3)

stairs([0;T],[U;U(end)])

ylabel('U (Ns/m/(1000kg))');

xlabel('time (m/s)');

axis([0 T(end),-1.1 1.1]);

figure(2)

plot(Z(:,1),Z(:,2))

xlabel('x (m)');

ylabel('v (m/s)');

axis([-5 105 -0.1 5]);

function f=boat\_obj(UDT)

Z0=[100;3];

U=UDT(1:4);

DT=UDT(5:8);

T=[DT(1);sum(DT(1:2));sum(DT(1:3));sum(DT)];

[t,Z]=ode45(@(t,Z) boat(t,Z,U(1)),[0,T(1)],Z0);

for i=2:4

[t,Z]=ode45(@(t,Z) boat(t,Z,U(i)),[T(i-1) T(i)],Z(end,:)');

end;

f=T(end)+10\*sum(Z(end,:).^2);

function dZdt=boat(t,Z,u)

b=0.05;

x=Z(1);

v=Z(2);

dxdt=-v;

dvdt=u-b\*v^2;

dZdt=[dxdt;dvdt];

***Other discussions:***

Although this calculation gives an ideal “optimal” control strategy for boat docking, in real life, this solution isn’t really practical. Because it is impossible to shift the engine from u=+1 to u=-1. Even if it is possible, it is still high risk to do so. In real boat docking problem, some other important factors should also be considered, such as security, energy efficiency, and so on.