MAE 6292 : Midterm Exam

03/27/2020

- Harshvardhan Uppaluru

Problem 1 :

$x \in \mathbb{R}^3$: location of the point

$$y_i = f_i(x) = \frac{1}{c_i^T x + d_i}(A_i x + b_i)$$

$A_i \in \mathbb{R}^{2\times 3}$ $b_i \in \mathbb{R}^{2\times 1}$ $c_i \in \mathbb{R}^{3\times 1}$ $d_i \in \mathbb{R}$ are sub-matrices of the camera matrix $P_i \in \mathbb{R}^{3\times 4}$

$$P_i = \begin{bmatrix} A_i & b_i \\ c_i^T & d_i \end{bmatrix}$$

Minimize the weighted sum of residual errors.

$$J(x) = \sum_{i=1}^{N} w_i \| y_i - f_i(x) \|^2$$

$w_i \in \mathbb{R}$ weighting parameters representing the degree of accuracy of i^{th} camera.

(a) Necessary conditions for optimality

$$0 = \frac{\partial J}{\partial x}\Big|_{x^*} = \frac{2w(Ad-bc)^T(b + Ax^* - dy - cx^*y)}{(cx+d)^3}$$

$$\frac{\partial^2 J}{\partial x^2}\Big|_{*} = \frac{2w(Ad-bc)(Ad-3bc+2c^2xy-2Acx+2cdy)}{(cx+d)^4} > 0$$

↖ positive semi-definite

$$\frac{\partial f}{\partial x} = \frac{Ad-bc}{(cx+d)^2} \times -2$$

(Attached the matlab file to derive the necessary conditions)

```
% Part(a) Necessary conditions for optimality
syms x y A b c d w
f = (A*x+b)/(c*x+d);
fx = simplify(jacobian(f,x))
```

fx =

$$\frac{A\,d-b\,c}{(d+c\,x)^2}$$

```
fy = simplify(jacobian(f,y))
```

fy = 0

```
J = sum(w * (norm(y - ((A*x+b)/(c*x+d))))^2);
Jx = simplify(jacobian(J,x))
```

Jx =

$$\frac{2\,w\,(A\,d-b\,c)\,(b+A\,x-d\,y-c\,x\,y)}{(d+c\,x)^3}$$

```
Jxx = simplify(jacobian(Jx,x))
```

Jxx =

$$\frac{2\,w\,(A\,d-b\,c)\,(A\,d-3\,b\,c+2\,c^2\,x\,y-2\,A\,c\,x+2\,c\,d\,y)}{(d+c\,x)^4}$$

```
Jy = simplify(jacobian(J,y))
```

Jy =

$$2\,w\left(y-\frac{b+A\,x}{d+c\,x}\right)$$

x is a vector; these are for a scalar x

```
% Part(b) Using Steepest Descent Method
clc;
clear all;
close all;
load("prob1_data.mat");

% Given Matrices
A = P(1:2,1:3,:);
b = P(1:2,4,:);
c = P(3,1:3,:);
d = P(3,4,:);

% Scalar constant
K = 0.001;

% Initial Guess
x = [5;5;5];
% Arrays for initial predicted_y and cost
pred_y = 0*ones(2,4);
Jx = [0;0;0];

for i = 1:4
    pred_y(:,i) = (A(:,:,i)*x + b(:,:,i))/(c(:,:,i)*x + d(:,:,i));
    % dJ/dx
    dJx(:,:,i) = (2*w(i)*(A(:,:,i)*d(:,:,i) - b(:,:,i)*c(:,:,i))'* ...
        (b(:,:,i) + A(:,:,i)*x - d(:,:,i)*y(:,i) - c(:,:,i)*x*y(:,i)))/ ...
        (d(:,:,i) + c(:,:,i)*x)^3;
    % This is to sum dJ/dx for the 4 camera
    Jx = Jx + dJx(:,:,i);
end

%fx = (A*d - b*c)/(d + c*x)^2;
%fy = 0;
%Jx = (2*w*(A*d - b*c)*(b + A*x - d*y - c*x*y))/(d + c*x)^3;
%Jy = 2*w*(y - (b + A*x)/(d + c*x));
% Looping variable
iter = 0;
while norm(Jx) > 1e-8
    % initial cost = 0
    J = 0;
    Jx = [0;0;0];
    for i = 1:4
        pred_y(:,i) = (A(:,:,i)*x + b(:,:,i))/(c(:,:,i)*x + d(:,:,i));
        dJx(:,:,i) = (2*w(i)*(A(:,:,i)*d(:,:,i) - b(:,:,i)*c(:,:,i))'* ...
            (b(:,:,i) + A(:,:,i)*x - d(:,:,i)*y(:,i) - c(:,:,i)*x*y(:,i)))/ ...
            (d(:,:,i) + c(:,:,i)*x)^3;
        % Update gradient vector
        Jx = Jx + dJx(:,:,i);
        J = J + (w(i)*(norm(y(:,i) - pred_y(:,i)))^2);
    end
    % Update x
    delta_x = K * Jx;
    x = x - delta_x;
```

```
    %delta_J = K*norm(Jx);
    %J = J - delta_J;
    iter = iter + 1;
end
% Part(c)
disp("Optimal Estimate")
```

```
Optimal Estimate
```

```
disp(x)
```

```
    5.1621
    4.8106
    5.2034
```

```
disp("Optimal cost")
```

```
Optimal cost
```

```
disp(J)
```

```
    0.0121
```

Problem 2: Trajectory optimization:

$y(0) = 0 \qquad y(x_f) = y_f$

Initial point : $(0, 0)$
Terminal point : (x_f, y_f)

(a) From conservation of Total Energy

$$K.E = P.E$$

$$\frac{1}{2}mv^2 = mgy$$

$$\boxed{v = \sqrt{2gy}}$$ ✓

(b) $ds = \sqrt{dx^2 + dy^2} \qquad v = \frac{ds}{dt}$

We have to minimize J to reach the terminal point in shortest time.

Total travel time $(T) = \int_0^{t_f} dt$

$$T = \int_0^{t_f} \frac{ds}{v} = \int_0^{t_f} \frac{\sqrt{dx^2 + dy^2}}{\sqrt{2gy}}$$

$$T = \int_0^{t_f} \sqrt{\frac{1 + (y')^2}{2gy}}\, dx \qquad \text{where } y' = \frac{dy}{dx}$$

∴ We can formulate the cost function as

$$\boxed{J(y, y') = \int_0^{t_f} \sqrt{\frac{1 + y'^2}{2gy}}\, dx}$$ ✓

(c) Let $L(y,y') = \sqrt{\frac{1+y'^2}{2gy}}$

Using Euler Lagrange differential equation:

$$\frac{\partial L(y,y')}{\partial y} - \frac{d}{dx}\left(\frac{\partial L(y,y')}{\partial y'}\right) = 0 \qquad -(1)$$

However in $L(y,y') \rightarrow x$ doesnot appear

Therefore

$$\frac{d}{dx}L(y,y') = \frac{\partial L(y,y')}{\partial y} \times \frac{\partial y}{\partial x} + \frac{\partial L(y,y')}{\partial y'} \times \frac{\partial y'}{\partial x}$$

$$\frac{d}{dx}L(y,y') = \frac{\partial L(y,y')}{\partial y} y' + \frac{\partial L(y,y')}{\partial y'} y''$$

Rearranging equations:

$$\frac{\partial L(y,y')}{\partial y} y' = \frac{dL(y,y')}{dx} - \frac{\partial L(y,y')}{\partial y'} y'' \qquad -(2)$$

multiplying (1) by y' and substituting (2) in (1), we get

$$\frac{dL(y,y')}{dx} - \frac{\partial L(y,y')}{\partial y'} y'' - \frac{d}{dx}\left(\frac{\partial L(y,y')}{\partial y'}\right) y' = 0$$

$$\frac{d}{dx}\left(L - \frac{\partial L}{\partial y'}(y,y')\, y'\right) - \frac{\partial L}{\partial y'}(y,y')\, y'' = 0$$

multiplying by (-) and rearranging, we get

$$\frac{\partial L}{\partial y'}(y,y')\, y'' + \frac{d}{dx}\left(y' \frac{\partial L}{\partial y'}(y,y') - L\right) = 0$$

Since $L_x = 0$, we get

$$\frac{d}{dx}\left(y' \frac{\partial L}{\partial y'}(y, y') - L\right) = 0$$

Let $\boxed{H(y, y') = y' \frac{\partial L}{\partial y'}(y, y') - L}$ ✓

$$H(y, y') = \frac{1}{\sqrt{(2gy)(1+y'^2)}}$$ (solved in the next sheet)

$\boxed{\frac{dH}{dt} = 0}$ $\therefore H(y, y')$ is preserved

(d) From (c) we have

$$\frac{d}{dx}\left(y' \frac{\partial L}{\partial y'}(y, y') - L \right) = 0 \quad \therefore L - y' \frac{\partial L}{\partial y'}(y, y') = K$$

for some constant of integration

Computing $\frac{\partial L}{\partial y'}(y, y')$, we have

$$\frac{\partial L}{\partial y'}(y, y') = y' \times \frac{1}{\sqrt{2gy}\sqrt{1+y'^2}}$$

Subtracting $y' \frac{\partial L}{\partial y'}(y, y')$ from L, we have

$$\sqrt{\frac{1+y'^2}{2gy}} - \frac{y'^2}{\sqrt{2gy}\sqrt{1+y'^2}} = K$$

Simplifying, we have

$$\frac{1}{\sqrt{2gy}\sqrt{1+y'^2}} = K$$

squaring both sides and rearranging

$$y(1+y'^2) = \frac{1}{2gK^2} = c$$

where $c = \frac{1}{2gK^2}$ = some positive constant

$$\boxed{y(1+y'^2) = c}$$

(e) Solving for y' in the previous equation, we get

$$y' = \sqrt{\frac{c-y}{y}}$$

Therefore $\frac{dy}{dx} = \sqrt{\frac{c-y}{y}}$ $\qquad$ $\frac{dx}{dy} = \sqrt{\frac{y}{c-y}}$

$$\frac{dx}{d\theta} = \frac{1}{2}c - \frac{1}{2}c\cdot\cos\theta$$

$$\frac{dy}{dx} = \frac{dy/d\theta}{dx/d\theta} = \frac{\sin\theta}{1-\cos\theta}$$

$$\frac{dy}{d\theta} = \frac{1}{2}c\sin\theta$$

Substituting $\frac{dy}{dx}$ and $\frac{dy}{d\theta}$ in $y(1+y'^2)$, we get

$$y(1+y^2)$$

$$\Rightarrow \left(\frac{1}{2}c - \frac{1}{2}c\cos\theta\right)\left(1 + \frac{\sin^2\theta}{(1-\cos\theta)^2}\right)$$

$$\Rightarrow \frac{1}{2}c\,(1-\cos\theta)\left(1 + \frac{\sin^2\theta}{(1-\cos\theta)^2}\right)$$

$$\Rightarrow \frac{1}{2}c\,(2) = c$$

Hence it is a solution of $y(1+y'^2) = c$

$$x(\theta) = \frac{1}{2}c(\theta - \sin\theta) \qquad y(\theta) = \frac{1}{2}c(1-\cos\theta)$$

✓

(9) Cycloid · Geometric meaning :

A cycloid is a curve that traces the path of a fixed point on a circle as the circle rolls along a straight line in 2-dimensions

* A cycloid through the origin, with a horizontal base given by the x-axis, generated by a circle of radius r rolling over the positive side of the horizontal base consists of pts

$$x = r(\theta - \sin\theta) \qquad y = r(1 - \cos\theta)$$

θ = angle through which the rolling circle is formed

```
% Problem 2(f)
clc;
close all;
clear all;

% symbolic variables
syms c theta

% We have to find c and theta_f
% Both the equations are defined here
eqns = [c*(theta - sin(theta))/2 == 1, c*(1 - cos(theta))/2 == 1];
vars = [theta c];
[theta_f, final_c] = solve(eqns, vars);
```

```
Warning: Unable to solve symbolically. Returning a numeric solution using vpasolve.
```

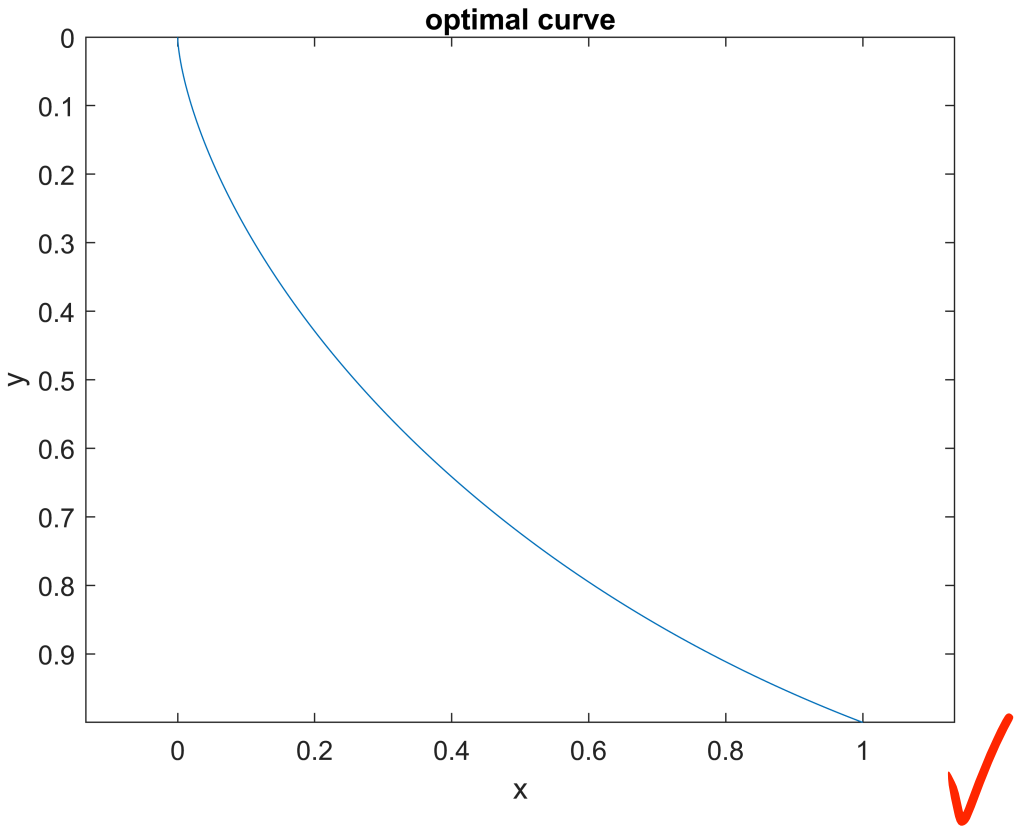
```
disp(final_c)
```

```
1.1458340750635006738963665178593
```

```
disp(theta_f)
```

```
2.4120111439135253425264820657 32
```

```
plt_theta = 0:0.001:theta_f;

% x and y represented in parametric form
x = final_c*(plt_theta - sin(plt_theta))/2;
y = final_c*(1-cos(plt_theta))/2;

% Plotting
figure;
plot(x,y)
title("optimal curve")
xlabel("x");
ylabel("y")
set(gca, 'Ydir','reverse');
axis equal;
```


optimal curve
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
y
0
0.2
0.4
0.6
0.8
1
x

Problem 3: Optimal control

constant speed = V control = steering angle = θ

Equations of motion: $\dot{x}_1 = V\cos\theta + c$ (effects of current), $\dot{x}_2 = V\sin\theta$ } f

$(x_1(0), x_2(0)) = (0,0)$

Shoreline: $S_f = \{x \in \mathbb{R}^2 \mid \|x-a\|^2 = r^2\}$ for $a=[a_1, a_2] \in \mathbb{R}^2$, $r>0$

(a)

$$J = \phi(t_f, x_f) + \left\{\int_{t_0}^{t_f} L(x,t,u)\,dt\right\}$$

in this case, goal is to minimize time, therefore we have

$$\Rightarrow \int_0^{t_f} 1\,dt = t_f$$

(L = 1)

Terminal constraint: $\psi(t_f, x_f) = S_f$

Hamiltonian:

$$H = L + \lambda^T f$$

$$H = 1 + \lambda_{10}(V\cos\theta + c) + \lambda_{20}(V\sin\theta)$$

λ = Lagrange multiplier for state

Augmented cost:

$$\bar{J} = \mu^T\psi + \int_{t_0}^{t_f} H - \lambda^T\dot{x}\,dt$$

μ = Lagrange multiplier for terminal state constraint

Final state must satisfy

$$\psi(x(t_f), t_f) = 0$$

Optimal conditions:

(1) $\dot{x} = \frac{\partial H^T}{\partial \lambda} = f$: state equation $\quad t \geq t_0$

$$f = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} v\cos\theta + c \\ v\sin\theta \end{bmatrix}$$

Size: (2×1)

(2) $\dot{\lambda} = -\frac{\partial H^T}{\partial x}$: co-state equation $\quad t \leq t_f$

$$\dot{\lambda}_{10} = -\frac{\partial H^T}{\partial x_1} = 0 \quad \Rightarrow \lambda_{10} = c_1$$

$$\dot{\lambda}_{20} = -\frac{\partial H^T}{\partial x_2} = 0 \quad \Rightarrow \lambda_{20} = c_2 \checkmark$$

$\Rightarrow$ constants

(b)

(3) control input = steering angle = unconstrained

$0 = \frac{\partial H}{\partial u}$: control equation (stationary condition)

$$-\lambda_{10} v\sin\theta^* + \lambda_{20} v\cos\theta^* = 0$$

$$\Rightarrow \tan\theta^* = \frac{\lambda_{20}}{\lambda_{10}}$$

[$\lambda_{20}, \lambda_{10}$ are constants $\therefore \theta^*(t)$ is a constant]

$$\boxed{\theta^* = \tan^{-1}\left(\frac{\lambda_{20}}{\lambda_{10}}\right)}$$

$\therefore$ optimal heading angle is fixed ✓

(c) Boundary conditions

$$x(t_0) = \text{given}$$

$$\left(\phi_x + \mu^T \psi_x - \lambda\right)^T\Big|_t dx(t) + \left(\phi_t + \mu^T \psi_t + H\right)\Big|_t dt = 0$$

Initial condition: $(x_1(0), x_2(0)) = (0,0)$

Optimal control is constant. Integrating equations of motion, we get

$$x_1 = vt\cos\theta + c \qquad x_2 = vt\sin\theta$$

Boundary condition: $(x_1(t_f), x_2(t_f))$ is a point on the circle S_f.

$$\psi = (x_1 - a_1)^2 + (x_2 - a_2)^2 = r^2$$

$$\left(\mu\psi_x - \lambda\right)\Big|_{t_f} = 0$$

$$\psi_x = \begin{bmatrix}\psi_{x_1}\\ \psi_{x_2}\end{bmatrix} = \begin{bmatrix}2(x_1 - a_1)\\ 2(x_2 - a_2)\end{bmatrix}$$ ✓

$$\mu\begin{bmatrix}2(x_1 - a_1)\\ 2(x_2 - a_2)\end{bmatrix} - \begin{bmatrix}c_1\\ c_2\end{bmatrix} = 0 \quad - (1)$$

$$\left(\mu\psi_t + H\right)\Big|_{t_f} = 0$$ ✓

Substitute x_1 & x_2 in ψ

$$\psi_{t_f} = 2(vt\cos\theta + c - a_1)v\cos\theta + 2(vt\cos\theta - a_2)v\sin\theta$$

$$\mu\left(2vt_f\cos\theta + c - a_1\right)v\cos\theta + 2\left(vt_f\cos\theta - a_2\right)v\sin\theta = 0$$ ✓

Problem 4: Optimal control

dynamics of a spacecraft: $\ddot{r} = \frac{-1}{r^3} r + u$

$\hookrightarrow r \in \mathbb{R}^2$ – position of spacecraft from center of earth

Equations of motion:

$$\dot{x}_1 = x_3 \qquad \dot{x}_3 = \frac{-1}{r^3} x_1 + u_1$$
$$\dot{x}_2 = x_4 \qquad \dot{x}_4 = \frac{-1}{r^3} x_2 + u_2$$

$$r = \sqrt{x_1^2 + x_2^2}$$

$$x_0 = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \qquad x_f = \begin{bmatrix} -2 & 0 & 0 & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

Cost:

$$J = \frac{1}{2} \int_0^{\pi} (u_1^2 + u_2^2) dt$$

(a)

Hamiltonian : $H(x, \lambda, u) =$

$$\frac{1}{2}(u_1^2 + u_2^2) + \lambda_1 x_3 + \lambda_2 x_4 + \lambda_3 \left(\frac{-1}{r^3} x_1 + u_1\right) + \lambda_4 \left(\frac{-1}{r^3} x_2 + u_2\right)$$

From $\left[\frac{\partial H}{\partial u} = 0\right]$, we get

$$\begin{bmatrix} u_1 + \lambda_3 \\ u_2 + \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow u_1 = -\lambda_3 \qquad u_2 = -\lambda_4$$

Substituting u_1, u_2 in the original Hamiltonian, we get

Reduced Hamiltonian:

$$H(x,\lambda) = \frac{1}{2}\left(\lambda_3^2 + \lambda_4^2\right) + \lambda_1 x_3 + \lambda_2 x_4 + \lambda_3\left(\frac{-1}{r^3}x_1 - \lambda_3\right) + \lambda_4\left(\frac{-1}{r^3}x_2 - \lambda_4\right)$$

we get

$$\boxed{H(x,\lambda) = \frac{-1}{2}\left(\lambda_3^2 + \lambda_4^2\right) + \lambda_1 x_3 + \lambda_2 x_4 - \lambda_3 \frac{1}{r^3} x_1 - \lambda_4 \frac{1}{r^3} x_2}$$

(c) Initial multiplier should be chosen such that terminal state satisfies $x(t_f) = x_f$

According to Quasi-linearization method, we have

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} \Phi_{xx} & \Phi_{x\lambda} \\ \Phi_{\lambda x} & \Phi_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} x(t_0) \\ \lambda(t_0) \end{bmatrix} + p(t)$$

Therefore we have

$$x(t) = \Phi_{xx}(t_f, t_0)\, x(t_0) + \Phi_{x\lambda}(t_f, t_0)\lambda(t_0) + p(t)$$

Replacing $t \to t_f$, we get

$$x(t_f) = \Phi_{xx}(t_f, t_0)\, x(t_0) + \Phi_{x\lambda}(t_f, t_0)\lambda(t_0) + p(t_f)$$

$$x_f - \Phi_{xx}(t_f, t_0)\, x_0 + p(t_f) = \Phi_{x\lambda}(t_f, t_0)\lambda(t_0)$$

$$\boxed{\lambda(t_0) = \Phi_{x\lambda}^{-1}(t_f, t_0)\left[x_f - \Phi_{xx}(t_f, t_0)\, x_0 - p(t_f)\right]}$$

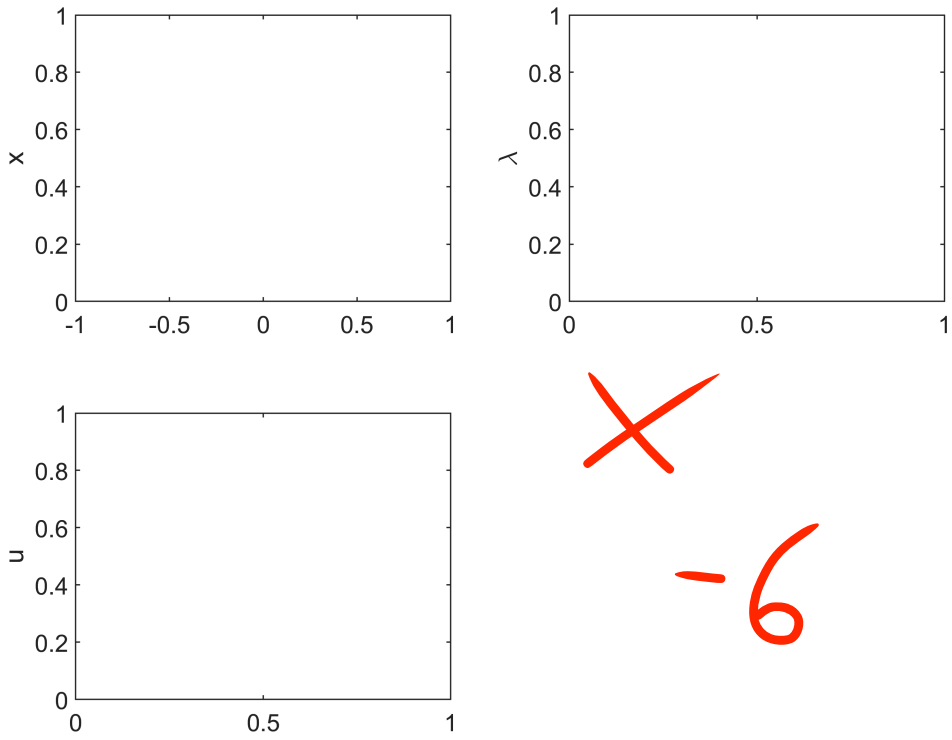
```
function [A, e] = eomLin(x, lambda)

    [Hx, Hl, Hxx, Hxl, Hlx, Hll]=prob4_H_deriv(x,lambda);
    A = [Hlx Hll;
        -Hxx -Hxl];
    e = -A*[x;lambda]+[Hl'; -Hx'];
end
```

```
%function prob4d
clc;
clear;
close all;

N = 1001;
t0 = 0;
tf = pi;
t = linspace(t0,tf,N);
dt = t(2)-t(1);

%
% init guess
x_i = 2*ones(N,4);
lambda_i = ones(N,4);
xf = [-2, 0, 0, -1/sqrt(2)];
x0 = [1, 0, 0, 1];
eps = 1e-6;
delta= 1.0;
x_i(1,:) = x0;

while delta > eps
    Phi = zeros(8,8,N);
    p = zeros(8,N);
    Phi(:,:,1) = eye(8);
    p(:,1) = [0,0,0,0,0,0,0,0]';
    for k=1:N-1
        [A,e]=eomLin(x_i(k,:)',lambda_i(k,:)');
        Phi_dot = A*Phi(:,:,k);
        p_dot = A*p(:,k) + e;
        Phi(:,:,k+1) = Phi(:,:,k) + Phi_dot*dt;
        p(:,k+1) = p(:,k) + p_dot*dt;
    end

    Phi_xl = Phi(1:4,5:8,N);
    Phi_xx = Phi(1:4,1:4,N);
    p_x = p(1:4,N);
    lambda0 = inv(Phi_xl) * (xf'-Phi_xx*x0'-p_x);

    x = ones(N,4);
    lambda = ones(N,4);
    x(1,:) = x0;
    lambda(1,:) = lambda0';
    z0 = [x0'; lambda0];
    for k = 2:N
        zk=Phi(:,:,k)*z0+p(:,k);
        x(k,:) = zk(1:4)';
        lambda(k,:) = zk(5:8)';
    end
    u = -lambda;


    % update x^i and lambda^i
```

```
    delta=max(abs(x-x_i))+max(abs(lambda-lambda_i));

    x_i=x;
    lambda_i=lambda;
end
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  1.496378e-20.
Warning: Matrix is singular, close to singular or badly scaled. Results may be inaccurate. RCOND = NaN.
```

```
figure;
subplot(2,2,1);
plot(t,x);hold on;
ylabel('x');
subplot(2,2,2);
plot(t,lambda);hold on;
ylabel('\lambda');
subplot(2,2,3);
plot(t,u);hold on;
ylabel('u');
```




```
%end
```