AMS 232 Nonlinear Optimization Hw #2

May 9, 2015

The following system describes the dynamics of an inverted pendulum where the moment of inertia of the pendulum and the friction between the pendulum and the cart are negligible. Defining x_1 as cart position, x_2 as cart velocity, x_3 as the angle between the pendulum and the positive vertical direction x_4 as the angular velocity, and u is the force applied to the cart, we have the dynamics of the system as

$$x_1 = x_2 \tag{1}$$

$$x_2 = \frac{mLx_4^2 \sin x_3 - mg \sin x_3 \cos x_3}{M + m\sin^2 x_3} + \frac{u}{M + m\sin^2 x_3} \tag{2}$$

$$x_3 = x_4 \tag{3}$$

$$x_4 = \frac{g(m+M)\sin x_3 - Lmx_4^2 \sin x_3}{LM + Lm\sin^2 x_3} - \frac{u\cos x_3}{LM + Lm\sin^2 x_3} \tag{4}$$

Where M, m, L are the mass of the cart, mass of the pendulum and length to the center of mass of the pendulum.

1. Linearize the inverted pendulum system about the equilibrium point $x^* = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ when given the input $u^* = 0$.

   We linearize about the equilibrium $x^* = 0, u^* = 0$ by $\dot{x} = Ax + Bu$ where $A = \left.\dfrac{\partial f}{\partial x}\right|_{x^*,u^*}$ is a 4×4 real matrix and $B = \left.\dfrac{\partial f}{\partial u}\right|_{x^*,u^*}$ is a 4×1 vector. As $f : \mathbb{R}^4 \longrightarrow \mathbb{R}^4$ and $u \in \mathbb{R}$ we have that

$$A = \left.\frac{\partial f}{\partial x}\right|_{x^*,u^*} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_4} \\ \vdots & & \vdots \\ \dfrac{\partial f_4}{\partial x_1} & \cdots & \dfrac{\partial f_4}{\partial x_4} \end{bmatrix}_{|x^*,u^*} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{gm}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(m+M)}{LM} & 0 \end{bmatrix}$$

$$B = \left.\frac{\partial f}{\partial u}\right|_{x^*,u^*} = \begin{bmatrix} \dfrac{\partial f_1}{\partial u} \\ \vdots \\ \dfrac{\partial f_4}{\partial u} \end{bmatrix}_{|x^*,u^*} = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{LM} \end{bmatrix}$$

□

Consider the infinite horizon optimal control problem:

$$\begin{aligned} \text{minimize} \quad & J[x(\cdot), u(\cdot)] = \frac{1}{2}\int_0^\infty x^T Q x + u^T R u \, dt \\ \text{subject to} \quad & \dot{x} = Ax + Bu \\ & x(t_0) = x^0 \in \mathbb{R}^4 \end{aligned}$$

Where $R = 1$, Q is a diagonal matrix with the main diagonal as $\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$. Note, for an infinite horizon LQR problem to be well defined, we need (A, B) controllable and $(A, Q^{1/2})$ to be observable, which is the case (Chen, Linear Control Text). Since A, B, Q, R are all time invariant, and that we are minimizing an infinite horizon problem, we have that the optimal feedback controller for this problem is given by the constant gain $K = -R^{-1}B^TP$ (so $u^* = Kx$) where P is the solution to the algebraic Riccati equation

$$PA + A^TP - P(BR^{-1}B^T)P + Q = 0$$

See M. Ross' text (A Primer on Pontryagin's Principle in Optimal Control), section 3.4 for a full reference.

2. Here we numerically solve the algebraic Riccati equation. To solve the algebraic Riccati equation, we implemented the method presented in the paper "A Schur Method for Solving Algebraic Riccati Equations" by Alan J. Laub. See Theorem 5, part 1 of the paper. Attached is some code to solve an algebraic Riccati equation; use it to validate the examples presented in the paper. With A, B, R, Q given after problem 1, the solution P to $PA+A^TP-P(BR^{-1}B^T)P+Q = 0$ is

$$P = \begin{bmatrix} 1.49043264127365 & 1.11069472908698 & 2.73480164461872 & 0.483208418726095 \\ 1.11069472908698 & 1.37635428641379 & 3.59282921982252 & 0.636471182115187 \\ 2.73480164461872 & 3.59282921982250 & 23.5303741786798 & 3.73957001770460 \\ 0.483208418726094 & 0.636471182115184 & 3.73957001770461 & 0.601161601327363 \end{bmatrix}$$

3. Substituting our feedback control $u^* = Kx = -R^{-1}B^TPx$ into the dynamics $\dot{x} = Ax + Bu$ we have $\dot{x} = (A - BR^{-1}B^TP)x$, which is a stable system, as every eigenvalue of the matrix $A - BR^{-1}B^TP$ has negative real part. Therefore as $t \longrightarrow \infty$, $x \longrightarrow 0$. The following figures are simulations of the closed loop performance of the linearized inverted pendulum, obtained by running the MatLab code attached at the end of this document.

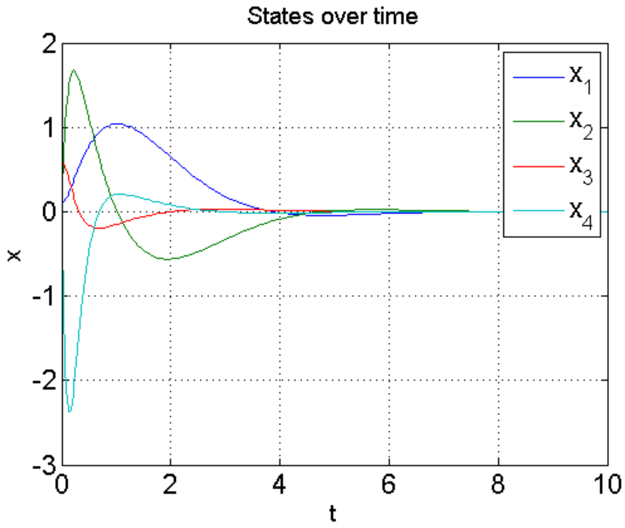


Figure 1: States over time given the initial condition $x_0 = \begin{bmatrix} 0.1 & 0.1 & 0.6 & 0.1 \end{bmatrix}$.

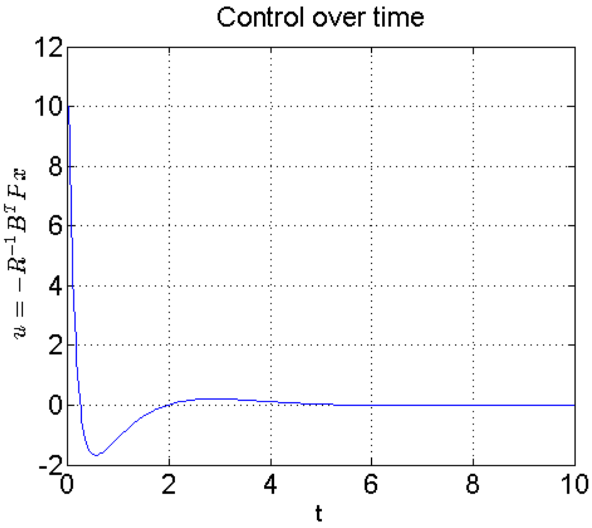
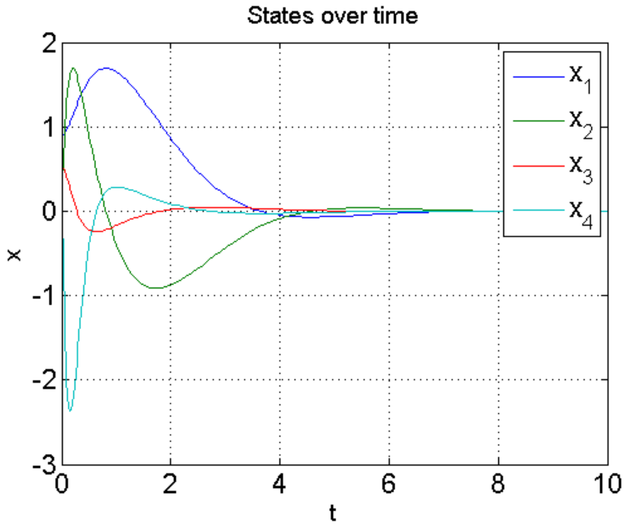


Figure 2: Control $u = Kx$ over time given the initial condition $x_0 = \begin{bmatrix} 0.1 & 0.1 & 0.6 & 0.1 \end{bmatrix}$.



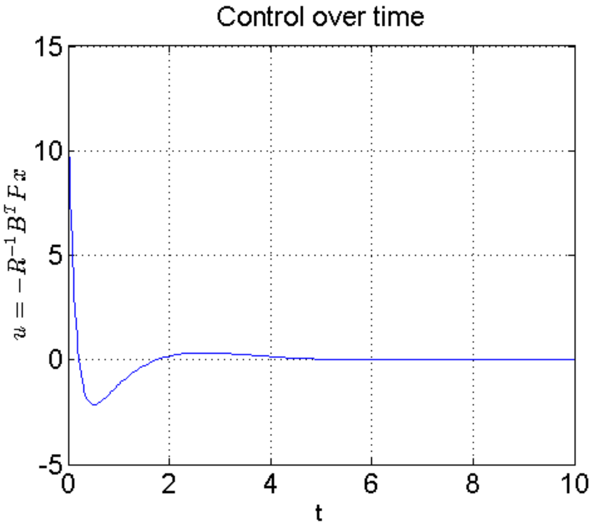
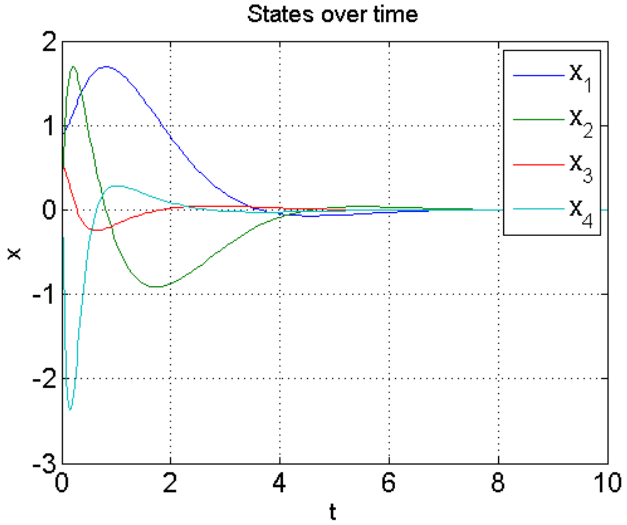


Figure 3: States over time given the initial condition $x_0 = \begin{bmatrix} 0.9 & 0 & 0.5 & 0.9 \end{bmatrix}$.

Figure 4: Control $u = Kx$ over time given the initial condition $x_0 = \begin{bmatrix} 0.9 & 0 & 0.5 & 0.9 \end{bmatrix}$.



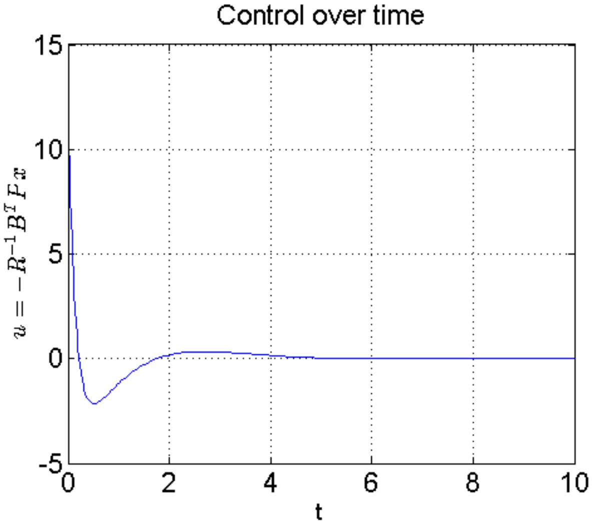


Figure 5: States over time given the initial condition $x_0 = \begin{bmatrix} 0 & 0.6 & 0 & 0.9 \end{bmatrix}$.

Figure 6: Control $u = Kx$ over time given the initial condition $x_0 = \begin{bmatrix} 0.1 & 0.1 & 0.6 & 0.1 \end{bmatrix}$.

4. Lastly, we analyze the effect of the closed loop performance to the inverted pendulum when varying R. As R is increased the time to reach the equilibrium increases.

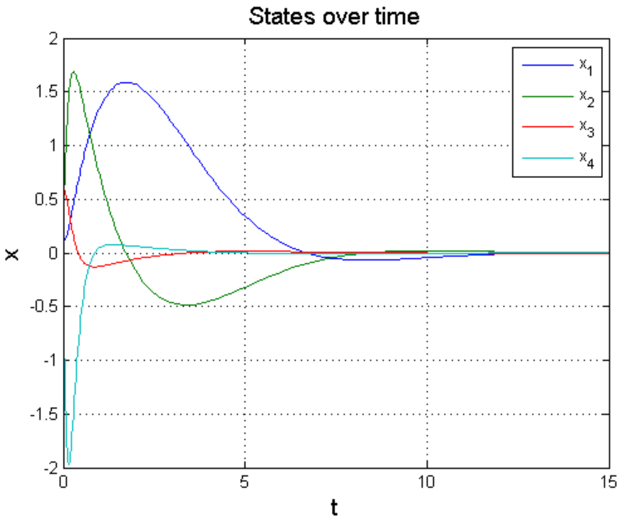


Figure 7: With $R = 10$, we see the states settle to the equilibrium around 11 seconds.

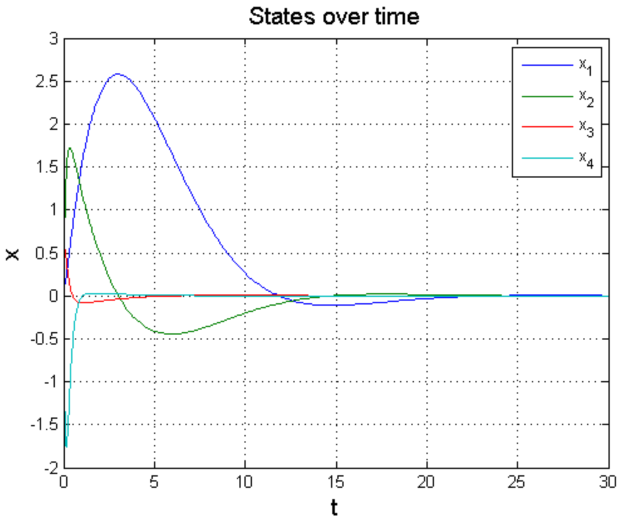


Figure 8: With $R = 100$, we see the states settle to the equilibrium around 23 seconds.

Matlab Code

hw2.m

```
%{
    Infinite Horizon LQR problem, HW #2 from AMS 232
%}
clear all;
close all;


%Simulation Parameters of the Inverted Pendulum
g = 9.8;    %m/s^2, downward is positive
m = 0.2;    %kg
M = 0.5;    %kg
L = 0.3;    %m

%A and B are the Linearized (Control) system
A = [0  1         0         0;
     0  0     -(g*m)/M      0;
     0  0         0         1;
     0  0  (g*(m+M))/(L*M)  0 ];
B = [0; 1/M; 0; -1/(L*M)];

%LQR simulation parameters
R = 1;
Q = diag([1 0 1 0]);

%The problem is an infinite horizon problem, so we set the final time to be large
t0=0;
tf=5;

x0 = [.1;.1;.6;.1];

%Example initial conditions
% ex1:  [.1;.1;.6;.1]
% ex2:  [.9;0;.5;.9]
% ex3:  [0;.6;0;.9]
% ex4:  [10;10;10;10] %the control for the linearized system fails for this large initCond

%Note, you should Check the Controllability of (A,B)
%Note, you should Check Observability of (A, sqrt(Q))
%If both tests are passed, then the problem is well defined

%Get P by solving the Continuous Time Algebraic Riccati Equation
P = solveContTimeAlgRiccati(A, B, Q, R);

% % We can check that indeed the CT Alg Riccati Equation is solved, just uncomment out the
% following two lines.
% checkResult = A'*P + P*A - P*( B*inv(R)*B' )*P + Q;
% norm(Result)

%The gain
K = -R*B'*P;

%Now we solve for the state x as \dot x = (A-B*inv(R)*B'*P)*x  the system is stable as
%all eigenvalues of (A-B*inv(R)*B'*P) have negative real part
linearSystem = @(t,x) (A-B*inv(R)*B'*P)*x;

%Just for fun, we get solve for x in the nonlinear system, passing-in the gain as the
%control.  For initial conditions near the equalibrium, both will result with the same
%state trajectories, but for large initial conditions, the linearization fails.
nonLinSystem =  @(t,x) ...
          [x(2);
          (m*L*x(4)^2*sin(x(3)) - m*g*sin(x(3)).*cos(x(3)))./(M+m*sin(x(3)).^2) + (K*x)./(M+m*sin(x(3)).^2);
          x(4);
          ((m+M)*g*sin(x(3)) - L*m*x(4)^2*sin(x(3)).*cos(x(3)))./(L*M + L*m*sin(x(3)).^2) - (K*x*cos(x(3)))./(L*M + L*m*sin(x(3)).^2)];

[tLin, xLinOpt] = ode45( linearSystem, [t0 tf], x0);
[tNonLin, xNonLin] = ode45( nonLinSystem, [t0 tf], x0);

%The optimal control is u=-R*B'*P*x where s is the solution
%to \dot x = (A-B*inv(R)*B'*P)*x
uOpt = K*xLinOpt';


%%%% Pretty Plot code %%%%
figure;
    plot(tLin, xLinOpt);
    title('States over time','fontSize',14);
    xlabel('t','fontSize',14);
    ylabel('x','fontSize',14);
    legend('x_1', 'x_2', 'x_3', 'x_4');
    grid on;

figure;
    plot(tLin, uOpt);
    title('Control over time');
    xlabel('t','fontSize',14);
    ylabel('$u=-R^{-1}B^TPx$','Interpreter','LaTex','fontSize',14);
    grid on;

%Plotting the state of the nonLinear system given the linearized control.
figure;
    plot(tNonLin, xNonLin);
    title('NonLinear System under the control u=Kx');
    grid on;
%
%   end hw2.m
%
```

solveContTimeAlgRiccati.m

```
%{
    This function solves the continuous time algebraic Riccati Equation, given
        A nxn
        B nx1
        Q nxn
        R 1x1
    We use the procedure from the paper ''A Schure Method for Solving Algebraic Riccati
    Equations'' by Alan J. Laub--See section II, theorem 5
%}
function [ P ] = solveContTimeAlgRiccati( A, B, Q, R )

    %A is a square matrix, so grab the number of rows n.
    [n,~] = size(A);

    %The Hamiltonian system  xDot      =  partial{H}/partial{lambda} and
    %                        lambdaDot = -partial{H}/partial{x}
    hamilSystem = [ A -(B*inv(R))*B' ; -Q -A'];

    %Perform a Schur Decomposition, and then reorder it so all of the negative eigenvalues
    %come first
    [U, T] = schur(hamilSystem);
    [US, TS] = ordschur(U,T,'lhp');

    %U11 and U12 match part (1) of Theorem 5
    U11 = US(1:n, 1:n);
    U21 = US((n+1):(2*n), 1:n);

    %result from part(1) of Theorem 5
    P = U21*inv(U11);

end

%
% end solveContTimeAlgRiccati.m
%
```