

SIGNALS AND SYSTEMS LABORATORY 8: State Variable Feedback Control Systems

INTRODUCTION

State variable descriptions for dynamical systems describe the evolution of the state vector, as a function of the state and the input. There may be an output equation as well, but we shall not be concerned with it in this discussion. For continuous time systems the state equation is a differential equation of the form

$$(1) \quad \dot{x}(t) = f(x(t), u(t)) .$$

In this equation, the state x and the control input u are vectors. Therefore the function $f(x, u)$ is also vector valued. If one makes the input a function of the state,

$$(2) \quad u(t) = g(x(t))$$

then the system is a state variable feedback system. The function $g(x)$ is called a *control law*. We shall study two examples of state variable control systems in this lab.

It is generally the case that the differential equations (1) are non-linear. Therefore they are very difficult to study, although they are not particularly difficult to simulate numerically using something like MATLAB. One analytical design tool which can be used in the vicinity of a vector x^0 is a *linearization* of the system. This is an approximation which is valid only for small distances from x^0 . Provided that $f(x^0, 0)$, or x^0 is a rest point, then the linearized system is

$$(3) \quad \dot{z} = Az + Bu ,$$

where

$$(4) \quad z(t) = x(t) - x^0$$

is the deviation of the state vector from the nominal point x^0 . The matrices A and B are computed from the function $f(x, u)$ as follows:

$$(5) \quad A_{i,j} = \left. \frac{\partial f_i}{\partial x_j} \right|_{x = x^0, u = 0}$$

and

$$(6) \quad B_{i,j} = \left. \frac{\partial f_i}{\partial u_j} \right|_{x = x^0, u = 0} .$$

The right hand side of equation (3) contains the linear terms in a Taylor series approximation of the right hand side of equation (1). Using the linearization, one can apply several well-known design methods for linear systems to obtain a control strategy. Once implemented however, the system will perform well only in the region for which the linear approximation is good. We will get a taste of this in the second problem to be studied in this lab.

TIME OPTIMAL CONTROL OF A LINEAR SYSTEM (BUSHAW'S PROBLEM)



Figure One A mechanical system acted upon by a force u .

Consider a rigid body of mass m acted upon by a single force. If the system is constrained to move in one dimension only then Newton's law of motion is $F = ma$. For simplicity, set the mass to 1, and call the force $u(t)$. Then the differential equation describing the motion is $\ddot{x} = u$. This system is the ultimate in simplicity, but it can make for an interesting control problem. In state variable form, it has dimension two. If we define the state vector to be

$$(7) \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix},$$

then the simple equation $\ddot{x} = u$ becomes

$$(8) \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot u,$$

which is in the form $\dot{x} = Ax + Bu$. One can make a feedback system out of this by taking the control input u to be a function of the state vector x . We shall make this problem more practical by constraining the input:

Control input constraint: $|u(t)| \leq 1$, for each t .

This constraint is realistic in many situations. For our problem, it limits the acceleration to at most 1. A typical problem involving a bounded input is the time optimal control problem. The goal is to drive the system from any initial state to a specified target in minimum time. The solution to this problem is usually *bang-bang*, which is a colorful way of saying that the control $u(t)$ is almost always 1 or -1. In other words the input is always on the boundary of the constrained set of allowable inputs. A feedback control law has the form

$$(9) \quad u = g(x_1, x_2).$$

If we respect the constraint, this function cannot be linear. Thus the feedback system will be *nonlinear*, and will be described by the differential equation

$$(10) \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} g(x_1, x_2).$$

Figure Two contains phase plane portraits of the nonlinear system described by equation (10), with different control laws. These are

$$u = 1$$

maximum acceleration

$$u = -1$$

maximum deceleration

$$u = \gamma(-x_1 - x_2), \text{ where } \gamma(v) = \begin{cases} 1, & \text{if } v \geq 1 \\ v, & \text{if } |v| \leq 1 \\ -1, & \text{if } v \leq -1 \end{cases}$$

linear with saturation

$$u = -\text{sign}(2x_1 + x_2|x_2|)$$

bang-bang control. u is always either full on or full off

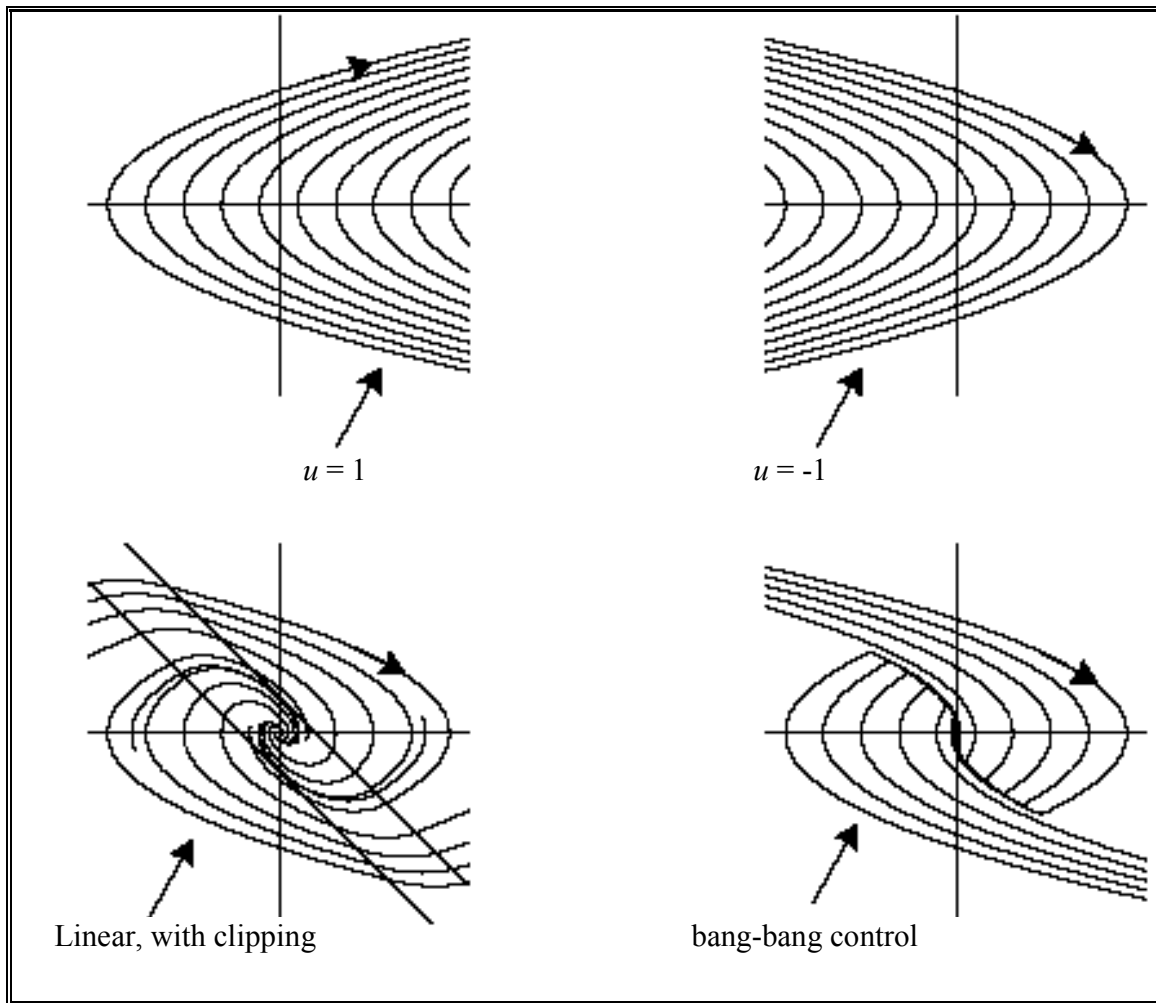


Figure Two Trajectories of the system $\ddot{x} = u$ with four choices for u .

Examine the m-file 'doublin.m', or *double integrator*, found on the web page, and make note of how the differential equations are numerically integrated, using MATLAB:

$$\begin{array}{llll} \dot{x}_1 = x_2 & \text{becomes} & x_1(t + dt) = x_1(t) + x_2 \cdot dt & \text{or} & x1=x1+dt*x2. \\ \dot{x}_2 = u & \text{becomes} & x_2(t + dt) = x_2(t) + u \cdot dt & \text{or} & x2=x2+dt*u. \end{array}$$

```
function doublin(a,b,g)
% doublin(a,b,g)
% run the double integrator.
% a and b are vectors of dimension 2
% g is a string describing the control function
% u=g(x1,x2)
L=1024;T=10;dt=T/L;t=dt*[0:L-1];epsilon=.1;
x=zeros(2,L);v=zeros(1,L);
x(:,1)=a;x1=a(1);x2=a(2);
k=2;
while (k<=L)&(sum(abs([x1;x2]-b))>epsilon);
    u=eval(g);
    x1=x1+dt*x2;x2=x2+dt*u;
    x(:,k)=[x1;x2];
    v(k)=u;
    k=k+1;
end
% We have not shown several lines which draw graphs
```

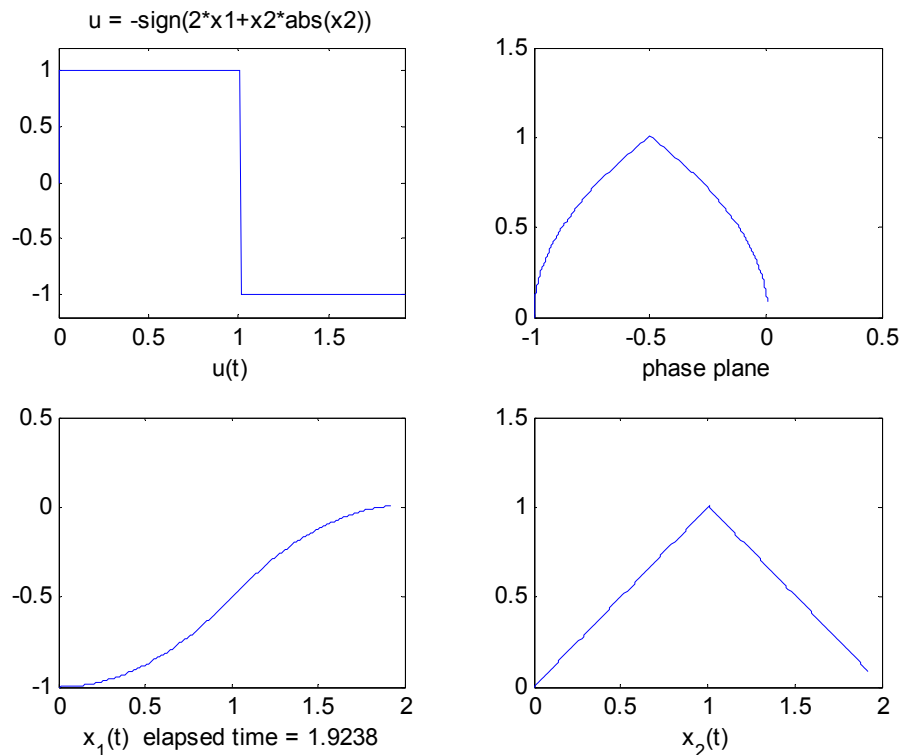


Figure Three A simulation of the system, using the bang-bang control law. The equations are $\dot{x}_1 = x_2$, $\dot{x}_2 = u$, which is apparent from the graph.

This tool numerically integrates the differential equations and produces the graphs shown in Figure Three. The control law is passed as a string, and therefore you can run the system for any law that you can write an expression for. The parameters **a** and **b** are column vectors of dimension 2, with **a** the initial state and **b** a target state. The simulation will terminate if the trajectory gets close to **b**, or if the time gets to 10, whichever happens first. Here is an example:

```
»g='-sign(2*x1+x2*abs(x2))';
»doublin([-1;0],[0;0],g)
1.9238
```

The final time is printed. In this example it is just short of 2 seconds. (See Figure Three.)

Assignment:

1. Compute the solutions to the differential equation (7) for arbitrary initial conditions, when $u = 1$. In other words find formulas for $x_1(t)$ and $x_2(t)$ when $x_1(0) = a_1$ and $x_2(0) = a_2$.
2. Design a control law $u = g(x_1, x_2)$ which takes the system to the point $x_1 = -3$, $x_2 = 0$ in minimum time. The control must not violate the constraint that $|u| \leq 1$, and should work for any choice of initial conditions. Use the tool 'doublin.m' to test your control law **g**. For example,

```
»doublin(randn(2,1),[-3;0],g)
```

exercises the system with a random initial state. For your report, include two randomized starts, and the specific case

```
»doublin([1;0],[-3;0],g)
```

Also report your control law **g**, and the design philosophy you used to get it.

THE CART AND THE PENDULUM

The *broom balancing* or *inverted pendulum* system shown in Figure Four has been a popular example for many years. The problem is to keep the ball in the air. Note that the applied force is rather indirect. We can only move the cart horizontally and thereby influence the pendulum indirectly. This problem is very close to the one that gave the early builders of rockets headaches. A rocket must be guided from the base by swiveling the rocket nozzles. If you have seen movies of early rocket launches, you will have seen rockets that had to be destroyed because they went off course. In this lab we will simulate the dynamics of this system using the MATLAB differential equation solver 'ode23'. We will simulate the motion with zero applied force, and then attempt to stabilize the system, about its unstable zero state position, by linearizing and then using a linear control law. You will study this stabilization problem, and get some data on its limitations.

The system is governed by a pair of coupled, second order nonlinear differential equations. In order to use the equation solver we shall put them in state variable form. Next, we will linearize the state variable equations about the unstable rest point where the cart and pendulum are at rest with the pendulum straight up: $x(t) = \dot{x}(t) = \theta(t) = \dot{\theta}(t) = 0$. For the resulting linear state variable equation, we can get a stabilizing linear feedback control law.

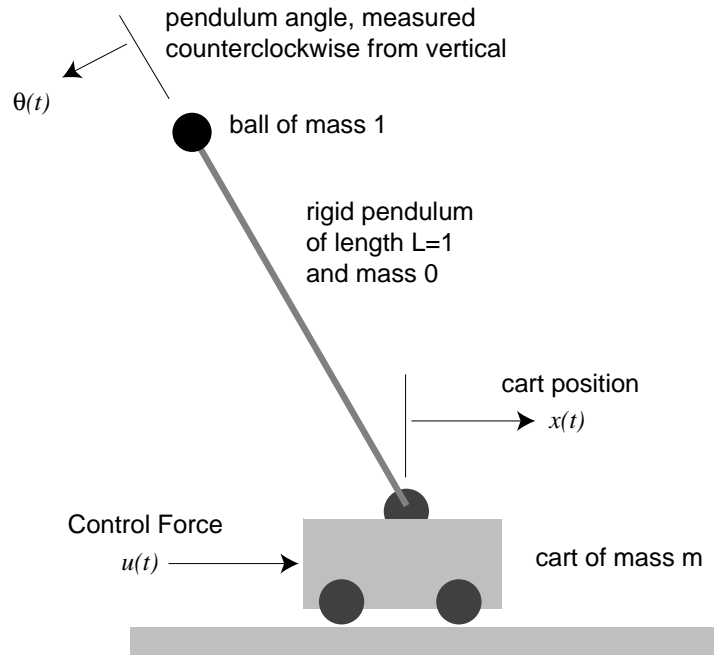


Figure Four *The Cart and the Pendulum* (with apologies to Edgar Allen Poe)

Here are all the equations:

Equations of motion of the cart and pendulum

$$(11) \quad \begin{aligned} (m+1)\ddot{x}(t) - \ddot{\theta}(t)\cos\theta(t) + \dot{\theta}^2(t)\sin\theta(t) &= u(t) \\ \ddot{\theta}(t) - \ddot{x}(t)\cos\theta(t) + \dot{x}(t)\dot{\theta}(t)\sin\theta(t) &= g\sin\theta(t) \end{aligned}$$

Definition of State vector for the cart and pendulum

$$(12) \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$$

Non-linear State Variable Equations for the cart and pendulum

$$(13) \quad \begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} m+1 & -\cos(x_2) \\ -\cos(x_2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} u - x_4^2 \sin(x_2) \\ (g - x_3 x_4) \sin(x_2) \end{bmatrix} \end{aligned}$$

Linearized State Variable Equations about the zero state

$$(14) \quad \dot{x} = Ax + Bu, \text{ where } A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{m}g & 0 & 0 \\ 0 & \frac{m+1}{m}g & 0 & 0 \end{bmatrix}, \text{ and } B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \\ \frac{1}{m} \end{bmatrix}.$$

Linear feedback control law

$$(15) \quad u = Gx, \text{ where } G = [g_1 \quad g_2 \quad g_3 \quad g_4].$$

Linearized Feedback system

$$(16) \quad \dot{x} = (A + BG)x.$$

The MATLAB differential equation solver 'ode23' may be used to compute solutions to the nonlinear equations (12). Use the 'help' facility to see how 'ode23' are used. To make the graph shown in Figure Five, use the following commands:

```
[t,y]=ode23('cart',0,10,[0;pi/6;0;0]);
subplot(2,2,1),plot(t,y(:,1));title('x(t)')
subplot(2,2,2),plot(t,y(:,2));title('\theta(t)')
```

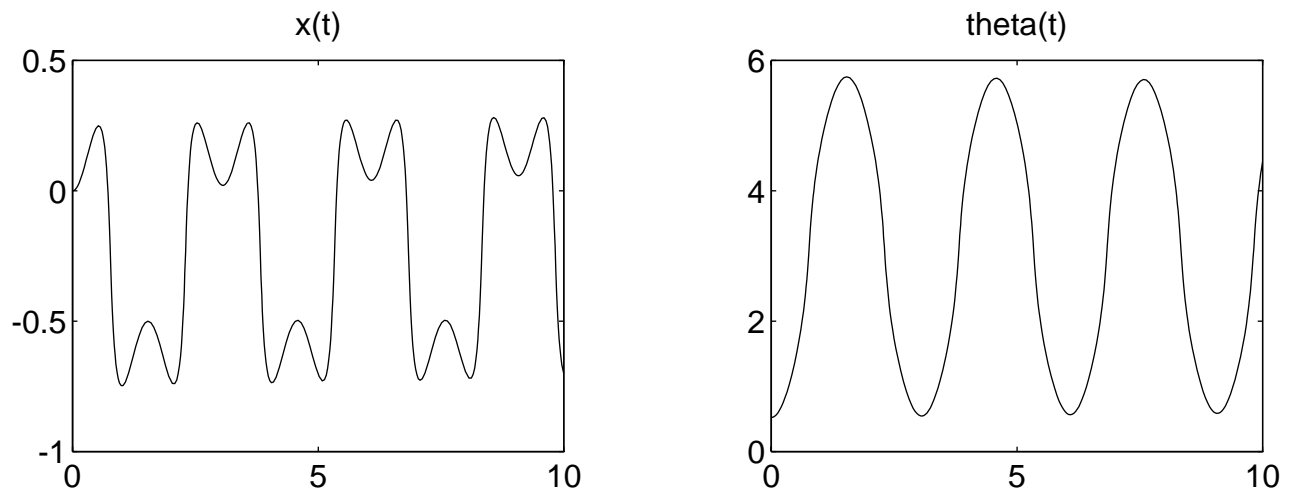


Figure Five Cart position and pendulum angle as a function of time, when the control force is zero, for initial conditions $x(0) = \dot{x}(0) = \dot{\theta}(0) = 0$, but $\theta(0) = \pi/6$. (This is the situation depicted in Figure Four.)

The string 'cart' identifies the m-file ['cart.m'](#) located on the webpage, which contains the equations.

```
function xdot=cart(t,x)
% xdot=cart(t,x)
% Nonlinear equations of motion for the Cart and
% pendulum system, for use by the MATLAB ode solver ode23

    g=10;M=1;u=0; %control force for free motion
    % u=[2.6 -53.6 5.4 -9.4]*x; % a linear feedback law
    c2=cos(x(2));s2=sin(x(2));
    xdot=[x(3:4);inv([M+1 -c2;-c2 1])*[u-x(4)^2*s2;(g-x(3)*x(4))*s2]];
```

Visualize the motion of the cart and pendulum as the curves in Figure Five depict it. The pendulum falls to the left and comes up again to an angle opposite the starting angle. It comes to rest momentarily and then reverses itself. Meanwhile the cart is rocking back and forth in a periodic motion. The horizontal component of momentum is zero for this set of initial conditions, and with no force applied. Therefore the cart does not have any aggregate horizontal motion. Notice that there is a choice of control inputs in the m-file ['cart.m'](#). You can activate the linear feedback law by removing the comment character.

With a little extra effort, one can even produce a primitive *movie* of the motion of the cart. The m-file ['cmovie.m'](#), found on the webpage under m-files for Lab 8, assembles several frames based on the solution to the differential equation, to make a movie.

```
»[t,y]=ode23('cart',[0 3],[0;pi/8;0;0]); % solve the diff. eqn.
»figure(1),plot(t,y(:,[1 2])) % This plots x(t) and theta(t)
»figure(2) % resize the window. Make it small!
»cmovie_demo % this uses the array y computed by ode23
»movie(M,3) % M is the movie produced by cmovie
```

Assignment:

3. Using the definitions in equations (5) and (6), compute the matrices A and B of the linearization of the system (12) about the zero state. Your results should agree with equation (14).
4. In MATLAB command mode, create the matrices A and B of the linearized equation (14), using the values $m=1$ and $g=10$. Then construct the 1 by 4 matrix $G=[2.6,-53.6,5.4,-9.4]$. Using the MATLAB 'eig' function to compute eigenvalues, compute the eigenvalues of the open-loop system, $\text{eig}(A)$, and the eigenvalues of the closed-loop system, $\text{eig}(A+B*G)$. What do you conclude about the behavior of the two systems in the vicinity of the zero state?
5. Now simulate the cart and pendulum system using 'ode23' in a manner similar to the example above, except that the m-file ['cart.m'](#) should be edited to activate the linear feedback control law. Use the same initial conditions as in the example, but with a greater time limit. Plot the resulting x and θ . Then try to make a movie. Now vary the initial conditions $[x_0 \theta_0 0 0]^T$. Making several runs, with a short time duration, try to characterize the region about the point $(0,0)$ in the $(x_1, x_2) = (x, \theta)$ plane for which the linear control system can balance the broom. Choose several pairs and then make marks on the plane of the form (+) if the pendulum is balanced and (-) if it isn't.