EE443 Lab 10: Simulated State Space Control of Inverted Pendulum

Introduction

State space control of the inverted pendulum depicted in figure 1 will be investigated through simulation with system parameters taken from the existing inverted pendulum in the laboratory. A controller will be designed for the inverted pendulum using state space techniques and verified through simulation. Next week the design will implemented on the physical system.

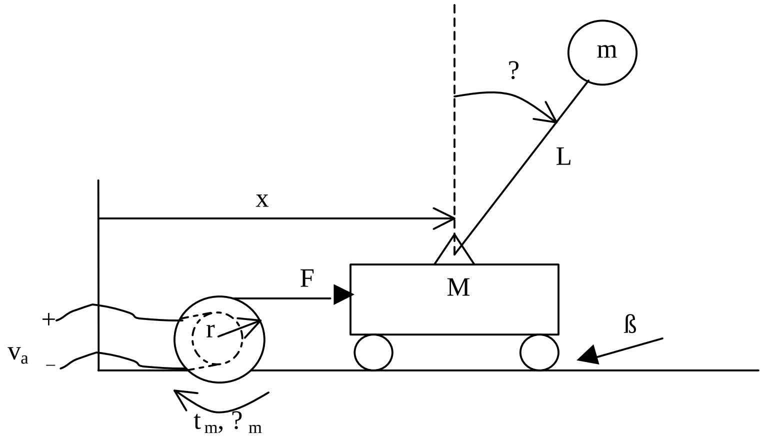

Figure 1: Inverted Pendulum

Procedure

The inverted pendulum shown in figure 1 can be modeled (with some effort) by the following two coupled differential equations. Throughout the derivation of these differential equations the pendulum angle *?* was assumed small, motor inductance *L* was assumed zero, and the pendulum mass *m* was assumed a point mass on the end of a massless shaft.

$$\ddot{q} - \left(\frac{b}{ML} + \frac{k_m k_b}{MLRr^2}\right)\dot{x} - \frac{(M+m)gq}{ML} = \frac{-k_m v_a}{MLRr}$$

$$\ddot{x} + \left(\frac{b}{M} + \frac{k_m k_b}{MRr^2}\right)\dot{x} + \frac{mgq}{M} = \frac{k_m v_a}{MRr}$$

Table 1: Parameters for the Inverted Pendulum	
PARAMETER	**DESCRIPTION**
r	effective pulley radius (m)
M	cart mass (kg)
m	pendulum mass (kg)
L	pendulum length (m)
k_m	motor torque constant (Nm/A)
k_b	motor back emf constant (V/rad/sec)
R	motor armature resistance (O)
ß	viscous friction of cart (N/m/sec)
g	gravity acceleration (9.81m/sec^2)
v_a	motor voltage (V)
F	force applied to cart (N)
x	cart position (m)
t_m	motor torque (Nm)
? $_m$	motor speed (rad/sec)
?	Pendulum angle from vertical (rad)

1. Write the coupled differential equations in state space form $\dot{\vec{x}} = A\vec{x} + Bu$, $\vec{y} = C\vec{x} + Du$ where $\vec{x} = \begin{bmatrix} x & \dot{x} & q & \dot{q} \end{bmatrix}^T$ are the system states, x, q are the system outputs, and the system input is $u = v_a$.
2. Determine all system parameter values from the existing system in the laboratory.
3. Enter your state space model into matlab and use the ctrb() and obsv() functions to construct the controllability and observability matrices. Is the system controllable and observable?
4. Define the state feedback controller $u = -K\vec{x} = -[k_1 \quad k_2 \quad k_3 \quad k_4]\vec{x}$ and apply it to your system using matlab's symbolic functions as shown below to find the characteristic equation in terms of the controller gains.

```
% A,B,C,D previously defined as numerical matrices
syms s k1 k2 k3 k4 K;          % create symbolic variables
K = [k1 k2 k3 k4];             % controller gains
char_poly = simple(det(s*eye(4,4)-(A-B*K))) % system char poly
```

5. Given the performance specifications of a settling time less than 2 sec and a damping ratio of approximately 0.7. Find four poles in general that will satisfy these criteria recalling that the design process will go nicely if two poles dominate the response. Use matlab's poly() and poly2str() functions to turn these poles into a characteristic polynomial.
6. Equate coefficients of the control system characteristic polynocial found in 4 with the coefficients of the characteristic polynomial containing the desired pole locations determined in 5 to find the corresponding controller gains.
7. Verify your controller gain calculations utilizing matlab's place() function.
8. Noting that the state space representation of the controlled inverted pendulum can be written as $\dot{\vec{x}} = (A - BK)\vec{x}$, $\vec{y} = C\vec{x} + Du$, use matlab's initial() function as shown below to simulate the controlled system for various initial conditions. Note that the gain matrix K should now be numerical. Verify that your design specifications were met and that motor voltages applied are practically achievable. Print these responses with appropriate labels including gains used.

```
% A,B,C,D,K are defined as numerical matrices/vectors
X0 = [-0.3; 0; 0.3; 0];          % initial configuration
[Y,X,T] = initial(A-B*K, zeros(size(B)), C, D, X0, 5);
subplot(2,2,1); plot(T,Y(:,1));  % cart position
subplot(2,2,2); plot(T,Y(:,2));  % pendulum angle
subplot(2,2,3); plot(T,-K*X');   % motor voltage
```

9. View the system's response with the inv_pend() function found on the network drive at n:\ee443\inv_pend.m and show this animation to the instructor.