DOE/OR/00033--T662

SWING-FREE TRANSPORT OF SUSPENDED LOADS

RECEIVED

JAN 23 1995

OSTI

SUMMER RESEARCH REPORT

SUBMITTED TO:

HBCU FACULTY/STUDENT RESEARCH PARTICIPATION PROGRAM

PREPARED BY:

A. M. HASANUL BASHER
ROBOTICS AND PROCESS SYSTEMS DIVISION
OAK RIDGE NATIONAL LABORATORY
OAK RIDGE, TENNESSEE

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

TABLE OF CONTENTS

Description	Page
ACKNOWLEDGMENTS	1
1. INTRODUCTION	2
2. SYSTEM DESCRIPTION	3
3. METHODOLOGY	3
4. DISCUSSIONS	4
5. APPENDIX	5
5.1 Simulation Plots	
5.2 Computer Program	

ACKNOWLEDGMENTS

This program is sponsored by Oak Ridge Institute for Science and Education (ORISE) at Oak Ridge, Tennessee, and conducted under the supervision of Dr. John F. Jansen in the Robotics and Process Systems Division (RPSD) of Oak Ridge National Laboratory. I wish to express my gratitude to my supervisor, John F. Jansen, for his invaluable suggestions and help in carrying out this project successfully.

I greatly appreciate ORISE, especially Deborah D. McCleary and Elizabeth M. Kittrell, for supporting me and providing an opportunity to work on this project. This project has broadened my horizon and enhanced my knowledge in the field of control theory. This opportunity helped to expose myself to an emerging technology, robotics, which is the technology of today and tomorrow and to get acquainted with the current research activities in the area.

My thanks are due for help in the computer laboratory provided by persons in RPSD.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

SWING-FREE TRANSPORT OF SUSPENDED LOADS

1. INTRODUCTION

Transportation of large objects using traditional bridge crane can induce pendulum motion (swing) of the object. In environments such as factory the energy contained in the swinging mass can be large and therefore attempts to move the mass onto target while still swinging can cause considerable damage. Oscillations must be damped or allowed to decay before the next process can take place. Stopping the swing can be accomplished by moving the bridge in a manner to counteract the swing which sometimes can be done by skilled operator, or by waiting for the swing to damp sufficiently that the object can be moved to the target without risk of damage.

One of the methods that can be utilized for oscillation suppression is input preshaping. The validity of this method depends on the exact knowledge of the system dynamics. This method can be modified to provide some degrees of robustness with respect to unknown dynamics but at the cost of the speed of transient response.

In this research we investigate and develop a controller using modern control theory to achieve oscillation damped movement of the suspended loads. The system contains uncertain parameters such as pendulum length and damping factor, and the sensor dynamics. The sensor is used to measure the angle of displacement of the pendulum. It is assumed that the uncertain parameters vary within known bounds. The controller is designed to ensure that the following characteristics are achieved:

- (a) The overall system is stable.
- (b) Oscillations die quickly.
- (c) The control input is smooth enough.

The controller to be designed utilizes the states of the system and those states must be made available to generate the control input. Only one of those states is available through the measurement of the angle of displacement of the pendulum. The other states are made available by designing an observer. So the task includes designing a controller and an observer. The performance of the system will deteriorate when an observer is added to the controller. We ensure that the system with controller-observer combined is stable. The gain matrices of the controller and the observer are found by using the pole placement technique.

The control algorithm developed in this research is tested through computer simulations using different parameter values. It is found that the derived controller holds the promise of achieving robustness to variations in system and sensor dynamics and produce damped oscillations within sufficiently small duration. It is also found that the system response at different parameter values within the specified range is superior when compared against that of from classical control method.

In section 2 the system under investigation is described. The methodology utilized in developing the control algorithm and computer simulation of the overall system is given in details in section 3. All the simulation results and the computer program developed are shown in the appendix.

2. SYSTEM DESCRIPTION

The transfer function of the plant is described as follows:

$$TF = \frac{k(b_3s^3 + b_2s^2 + b_1s)}{a_5s^5 + a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0}$$

where the coefficients a's and b's are functions of pendulum length (l), pendulum damping (d), and sensor natural frequencies (w_{n1} and w_{n2}).

The uncertain parameters can vary within the following known bounds:

$$\begin{aligned} 80 &\leq l \leq 60 \text{ ft} \\ 0.6 &\leq d \leq 0.1 \\ 4.2 &\leq w_{n1} \leq 2.8 \text{ Hz} \\ 1.536 &\leq w_{n2} \leq 1.024 \text{ Hz} \end{aligned}$$

The above transfer function includes the vector drive to move the crane, sensor dynamics and the dynamics of the pendulum. The objective is to design a suitable controller to produce swing-free movements of the pendulum.

3. METHODOLOGY

The complete task is divided into two phases such as i) Designing the controller and observer, ii) Testing the controller and observer connected to the system through computer simulations.

DESIGN:

The design steps are summarized as follows:

- The nominal values for the uncertain parameters are determined and then the nominal system matrices (A,B,C and D matrices) are obtained.

- The uncertain matrices are determined.
- The controller gain matrices are selected using pole placement technique.
- Observer system matrices are found from the plant nominal matrices and utilizing pole placement technique.
- The control input is derived next.
- The overall dynamics are determined with controller only and with controller-observer combined.

COMPUTER SIMULATION:

The following paragraph describes the steps taken to verify the algorithm through computer simulation.

- MATLAB software package is used for system simulation.
- Command input is generated that is being used as an input signal (test signal).
- The system is simulated with controller only, the system performances are observed and plots are obtained.
- The system is simulated with controller-observer combined, the performances are observed and plots are obtained.
- The control input and the pendulum states, especially the pendulum position, are generated at different parameter values using trapezoidal rule for integration.

4. DISCUSSIONS

The primary purpose of the project is to design a controller which is capable of generating smooth control input signal for the plant to produce swing-free motion of the suspended object driven by overhead crane. The controller designed for this purpose has resulted in an overall system whose performance is very satisfactory even when the uncertain parameters of the plant are at the worst situation. The controller needs the system states to generate control signals and to make those states available an observer is also necessary. This research develops an observer for the system also. It is noticed that the inclusion of the observer in the system deteriorates the overall performance. The performance of the system with controller-observer combined is still superior compared to other methods. The overall result of this investigation is very satisfactory. This control algorithm is verified via computer simulation. This algorithm can be tested in real system in RPSD.

The system performance deteriorates i.e. it takes longer time for the oscillation to dampen if the length of the pendulum is increased or the sensor natural frequency w_{n1} is increased or the sensor natural frequency w_{n2} is decreased. This can be seen from the plots for pendulum position. It is also noticed that as long as the uncertain parameters lie within the assumed bounds the oscillation dies within 50 seconds in the worst case.

5. APPENDIX

length = 60', damping = 0.5, $\omega_{n1} = 4.2$ and $\omega_{n2} = 1.03$

Figure 1: With Controller only

length = 60', damping = 0.5, $\omega_{n1} = 4.2$ and $\omega_{n2} = 1.03$

Figure 2: With Controller-Observer combined

length = 80', damping = 0.5, $\omega_{n1} = 4.2$ and $\omega_{n2} = 1.03$

Figure 3: With Controller only

length = 80', damping = 0.5, $\omega_{n1} = 4.2$ and $\omega_{n2} = 1.03$

Figure 4: With Controller-Observer combined

length = 60', damping = 0.3, $\omega_{n1} = 3.5$ and $\omega_{n2} = 1.28$

Figure 5: With Controller-Observer combined

length = 80', damping = 0.3, $\omega_{n1} = 4.2$ and $\omega_{n2} = 1.03$

Figure 6: With Controller-Observer combined

```

% calculate constants
lmc = 80;
lmu = 60;
dmpmc = 0.6;
dmpmn = 0.0;
% maxon damping
% minn damping
% calculate constants
kenn = 1.1; kennn = 0.9;
wml = 2*pi*w1lnh;
wmlmc = 2*pi*w1lnhc; wmlmn = 2*pi*w1lnhm;
wmlz = 2*pi*w2lnh; wmlzmc = 2*pi*w2lnhc; wmlzmn = 2*pi*w2lnhmn;
wmp = 2*pi*wmpn; wmpz = 2*pi*wmpzn;
pd = 2*pi*pdn;
kd = kdms*100/(2.54*12);
g = 32.2;

% of system transfer function
% Calculate coefficients: numerator and denominator
kt = -ke*kd*pd*w2.^2/w1.^2;
ktmn = -kenn*kd*pd*w2mn.^2/w1mn.^2;
kcn = kt/kcn*^(1/lmu);
numr = kt*(1/lp);
numrkc = kcn*^(1/lmn);
numrnm = kcnm*^(1/lmxc);
ap(5) = 2*dmp*wn2+pd;
aam(5) = 2*dmpmc*wn2mc+pd;
aap(4) = wn2.^2*g/lp+2*dmp*wn2;
aam(4) = wn2mn.^2*g/lmn+2*dmpmn*wn2mn;
aap(3) = 2*dmp*pd*^(g/lp)*wn2+wn2.^2*pd+g*pd/lp;
aam(3) = 2*dmpmn*pd*^(g/lmn)*wn2mn+wn2mn.^2*pd+g*pd/lmn;
aap(2) = g*wn2.^2/lp+2*dmp*^(g/lp)*wn2*pd;
aam(2) = g*wn2mn.^2/lmn+2*dmpmn*^(g/lmn)*wn2mn*pd;
aap(1) = g*wn2.^2*pd/lp;
aam(1) = g*wn2mn.^2*pd/lmn;
% model reduction (one state reduction)
a5 = (aam*(5)+aamn(5))/2; ax5=a5; an5=a5;
numn = [numr(1)/a5 0 0 0];
aap/a5;
nuden = [1 aap(4) aap(3) aap(2) aap(1)];
numnm = [1 aapc(1)/ax5 0 0 0];
aamc/a5;
ndenumn = [1 aamc(4) aamc(3) aamc(2) aamc(1)];
numnn = [1 aamnn(1)/an5 0 0 0];
aamnc/a5;
ndennn = [1 aamnn(4) aamnn(3) aamnn(2) aamnn(1)];

```

```
% form a,b,c,d matrices and uncertain matrices
[ap,bpp,dpp] = tf2ss(numu,nden);
ap = rot90(ap,2); bp=rot90(bpp,2); dp=rot90(dpp,2);
[amxt,bmxt,cmxt,dmxt] = tf2ss(numux,ndenux);
aimx=rot90(amxt,2); blmx=rot90(bmxt,2);
clmx=rot90(cmxt,2); dlmx=rot90(dmxt,2);
[amnt,bmnt,cmnt,dmnt] = tf2ss(numnu,ndennu);
almn=rot90(amnt,2); blmn=rot90(bmnt,2); clmn=rot90(cmnt,2);
dmn=rot90(dmnt,2);
an = (aimx+almn)/2; bn = (blmx+blmn)/2;
cn = (clmx+clmn)/2; dn = (dlmx+dlmn)/2;
delta = ap-an; deltb = bp-bn; deltc = cp-cn;

% gain select, pole placement technique
ppoles = [-1 -2 -3 -4]; obpoles = [-1 -2 -3 -4];
kc = acker(an,bn,ppoles);
as = an; cs = 2*cn;
l = acker(as',cs',obpoles);

% observer test nominal & uncertain system
delb is zero
zm = [0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
asxz = [an-bn*kc;l*cp an-l*cn-bn*kc];
asxu = [ap-bp*kc;l*cp an-l*cn-bn*kc-l*delc an-l*cn];
% transfer function (controller-observer combined)
aobs = an-bn*kc-l*cn;bobs=l;cobbs=kc;dobs=[0];
[nummob,denob] = ss2tf(aobs,bobs,cobs,dobs);
[zt,pt,gt] = ss2zp(al,bt,ct,dto);
[al,bt,ct,dto] = feedback(ap,bp,cp,dp,aobs,bobs,cobs,dobs);
[numnt,dennt] = ss2tf(al,bt,ct,dto);
[zt,pt,gt] = ss2zp(al,bt,ct,dto);

% root locus
dl = [0]; % state feedback, no observer
% pause
ak = an-bn*kc-l*cn; bk = l; ck = kc; dk = [0];
else
    lu = lu-2*dt;
elseif t > 1
    lu = lu+2*dt;
if t < 0.5
    if t < 1.5
        % generate command input
        uslope = [uslope (uxz(j)-uxz(j-1))/dt];
        ul = -kcxz*xz(:,j); uxz = [uxz ul];
        j = j+1;
        xx = [xz xz3];
        xz3 = xz(:,j)+(dt/2)*lxz2;
        lxz2 = apxz*lxz1+bpxz*(uxz(j)+apxz*xz(:,j)+bpxz*(uxz(j)+lu);
        lxz1 = xz(:,j)+dt*(apxz*xz(:,j)+bpxz*(uxz(j)+lu);
        for t = dt:dt:cmax
            % trapezoidal rule for integration
            uslope = 0;
            out = cpxz*xz(:,j)+dpxz*(uxz+lu);
            ul = -kcxz*xz(:,j); uxz = ul;
            kcxz = [0 0 0 kc];
            j = 1;
            dpxz = zpxz;
            bpxz = [bp;bn+l*dp]; cpxz = [cp 0 0 0];
            apxz = [apx axz;azx azz];
            axz = [0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
            axc = ap; azz = an-l*cn; axx=l*cp;
            xz = [0 0 0 0 0 0];
            u = 0; lu = 0; comp = 0; uz = 0;
            % simulation controller-observer combined
            if simul == 2;
                % simulation
                pause
                magin(mag,phase,w)
                [mag,phase,w] = bode(agk,bgk,cgk,dgk);
                pause(10)
                bode(agk,bgk,cgk,dgk);
                pause
                % bode plot
                [k,pp] = rlocfind(agk,bgk,cgk,dgk);
                xlocus(agk,bgk,cgk,dgk);
                [agk,bgk,cgk,dgk] = series(ap,bp,cp,dp,ak,bk,ck,dk);
```

```

        iu = 1;
    end
else
    iu = 0;
end
cominp = [cominp iu];
out = [out cp*x(:,j)+dp*(uxz(j)+iu)];
end

t = 0:dt:tmax;
subplot(221), plot(t,xz(1,:),t,xz(5:)), title('state 1')
subplot(222), plot(t,xz(2,:),t,xz(6:)), title('state 2')
subplot(223), plot(t,xz(3,:),t,xz(7:)), title('state 3')
subplot(224), plot(t,xz(4,:),t,xz(8:)), title('state 4')
pause(10)
subplot(221), plot(t,cominp(:)), title('command input')
subplot(222), plot(t,uxz(:)), title('control input')
subplot(223), plot(t,out(:)), title('pendulum position')

else
% Simulation with controller only

u = 0; iu = 0; cominp = 0;
x = [0 0 0 0]';
j = 1;
u = -kc*x(:,j);
out = cp*x(:,j)+dp*(u+iu);

for t = dt:dt:tmax

    l1 = x(:,j)+dt*(ap*x(:,j)+bp*(u(j)+iu));
    l2 = ap*l1+bp*(u(j)+iu)+ap*x(:,j)+bp*(u(j)+iu);
    l3 = x(:,j)+(dt/2)*l2;
    x = [x l3];

    j = j+1;

    ul = -kc*x(:,j);
    u = [u ul];

% generate command input

    if t < 1.5
        if t < 0.5
            iu = iu+2*dt;
        elseif t > 1
            iu = iu-2*dt;
        else
            iu = 1;
        end
    else
        iu = 0;
    end

    cominp = [cominp iu];
    out = [out cp*x(:,j)+dp*(u(j)+iu)];
end

end
% plot

t = 0:dt:tmax;

subplot(221), title('pendulum position ')
subplot(222), title('control input')
subplot(223), title('command input ')
subplot(224), title('states x1,x2,x3,x4')
plot(t,x(1,:),t,x(2,:),t,x(3,:),t,x(4:)),
xlabel('length = 60 ft, damping = 0.5')

end
print outpl62 -dps

```