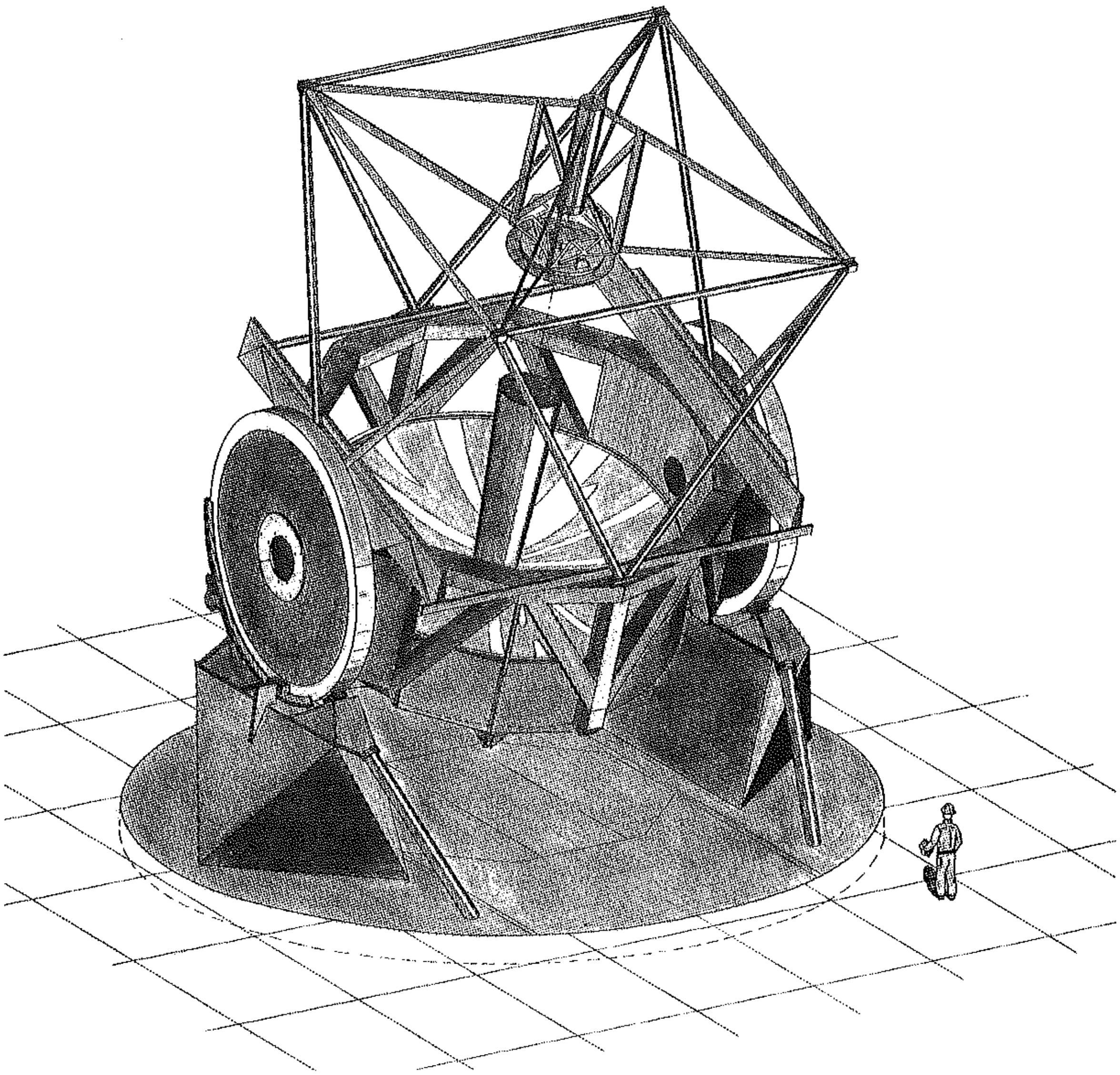
MAGELLAN PROJECT

University of Arizona Carnegie Institution of Washington The Johns Hopkins University



Choosing the Right Actuator for a Steering Secondary Mirror

J. Alan Schier
The Observatories of the
Carnegie Institution of Washington
Pasadena, California
November 1991
No. 34

INTRODUCTION

In a mechanical servo system, the type of actuator selected will have a strong effect on the system dynamics and the servo performance. This report examines a specific case in which a secondary mirror is servo controlled. We consider the use of two types of actuators: one inherently stiff, such as a piezoelectric device, and the other inherently soft, such as a voice coil. The results show that there is a bandwidth limit introduced by a stiff actuator that can be avoided by the use of a soft actuator.

DYNAMIC MODEL

The model in Figure 1 is used to generate the analytical results. It embodies the essential dynamics involved in controlling a secondary mirror suspended from a flexible support structure, which in turn is supported by the relatively massive center section of the telescope.

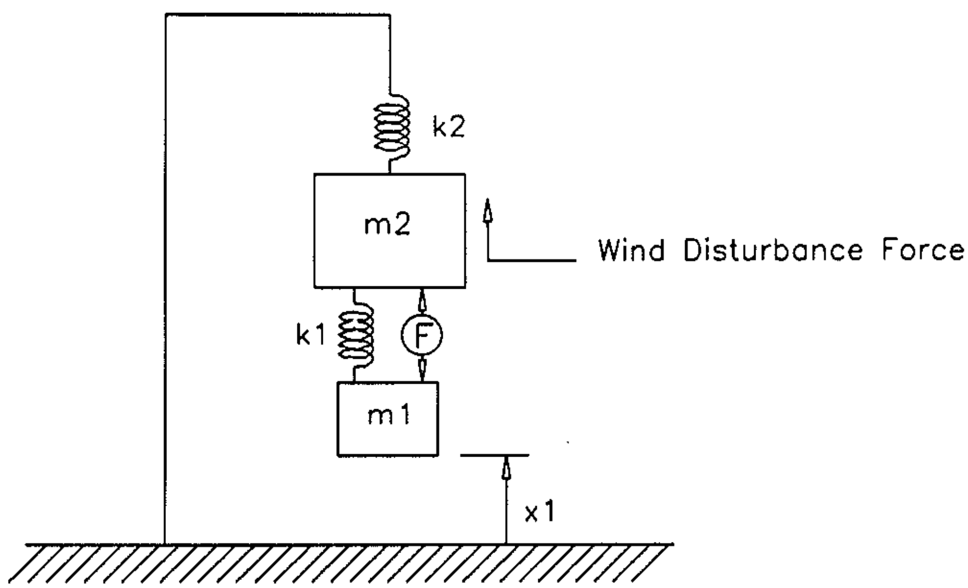


Figure 1.
Model of secondary support and control used to generate the analytical results of the report.

This model is of course a simplified version of the physical system. The important approximations are the following:

> The massive center section is modeled as ground since its inertia is orders of magnitude larger than any other inertia under consideration.
>
> The resonant support structure is modeled as having one significant mode of vibration, m_2 in combination with k_2. Other modes would not affect the qualitative

results of this analysis.

The mirror is represented by a single mass, m_1, controlled in translation. Since the angular motion of a controlled secondary mirror is quite small, the linear (small angle) approximation for this inertia is sufficiently accurate. Roughly speaking, the single mass in the model is that fraction of the secondary mirror supported by one actuator.

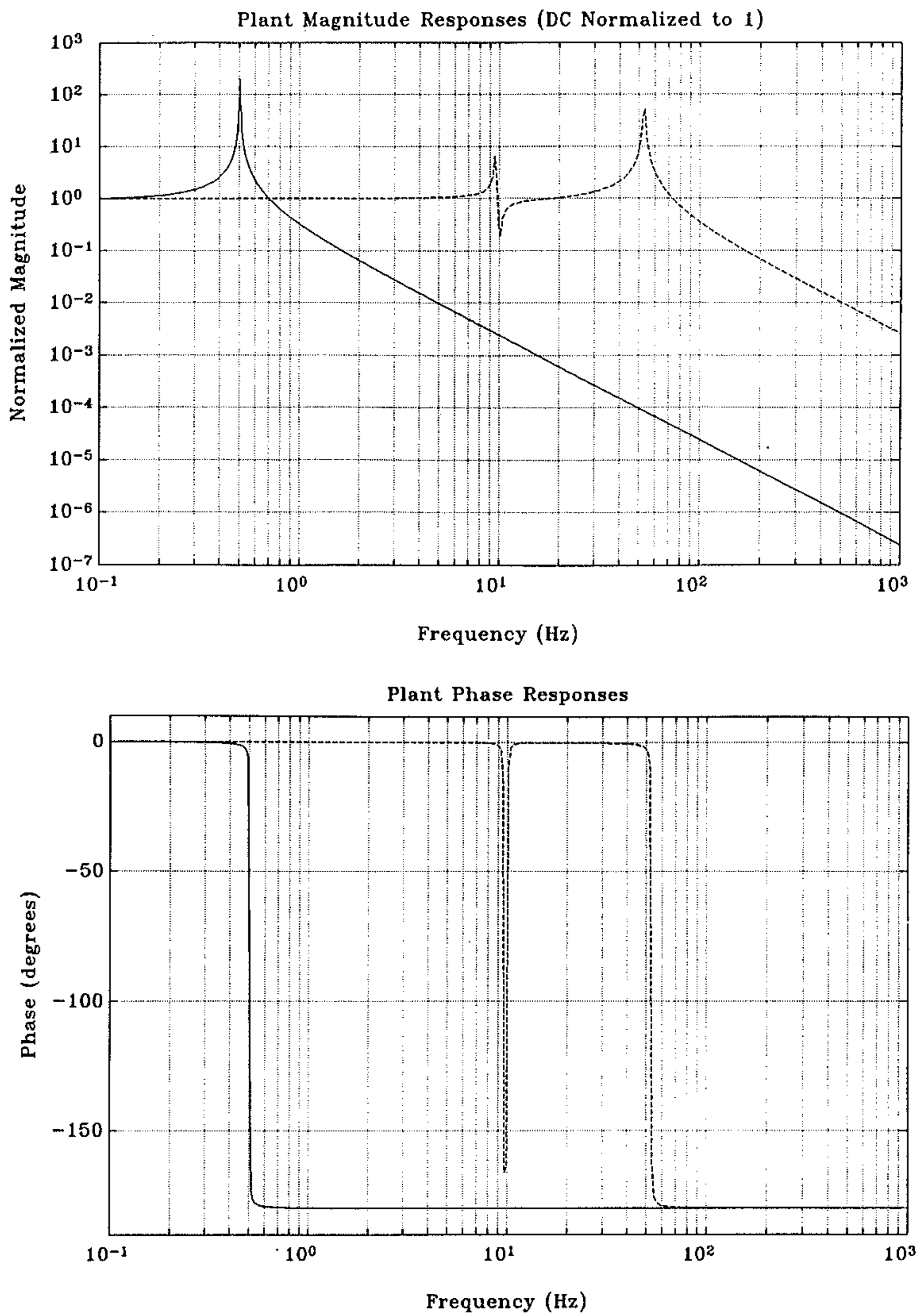
The actuator is modeled as a spring, k_1, in parallel with a source of an actuating force. The spring is assigned one of two stiffnesses, depending on whether a stiff or soft actuator is under consideration. The actuator then provides a force that moves the mirror mass while extending or compressing k_1. The position of m_1 is measured relative to inertial ground (or equivalently in this case, the telescope center section). This is essentially the measurement provided by a guide camera.

For the purposes of this report, a servo controller was designed to suit each of the two types of actuators with the constraint that they be of comparable complexity; otherwise, it would be possible to artificially favor one actuator over the other. The equations of motion for the model plant are given in Appendix A, and the controller equations are in Appendix B. The performance of the actuators with their controllers are evaluated for rejection of a wind disturbance applied at m_2 and for response to an input transient.

RESULTS

Figure 2a shows the magnitude responses for the plant with a stiff and soft actuator. Aside from the low-frequency resonance at 0.5 Hz, the soft-actuator case is relatively featureless. On the other hand, the stiff-actuator case shows a resonance-antiresonance pair near 10 Hz, and this is a bandwidth limiting feature. The phase responses in Figure 2b confirm these observations, with the soft actuator having one negative step in the phase and the stiff actuator showing a sharp negative spike near 10 Hz. It is really this negative spike combined with the resonance peak in the magnitude response that is troublesome.

In Figure 2a, the mode shape corresponding to the 0.5 Hz resonance peak is dominated by motion of the secondary mirror against the compliance of the soft actuator (m_1 and the soft k_1). At frequencies above this resonance, the secondary is decoupled from the support structure, and the system output behaves like a rigid body. For the resonance near 50 Hz, the mode shape is again almost entirely motion of the secondary mirror, but this time it is working against the stiff actuator (m_1 and the stiff k_1); the secondary is again decoupled from the structure above this resonance. The mode shape of the antiresonance near 10 Hz is primarily the resonant mode of the secondary support alone (m_2 against k_2). The resonance peak just below it is very nearly the combined masses of the secondary mirror and support structure on the support structure compliance (m_1 and m_2 on k_2). The actual parameters used to calculate these responses are at the beginning of the program listing in Appendix C.



Figures 2a,b.
Plant magnitude and phase responses for the soft (solid line) and stiff (dashed line) actuator cases. The resonance-antiresonance pair near 10 Hz in the stiff actuator case is particularly important since it is a bandwidth limiting feature. 10 Hz corresponds roughly to the resonant mode of the support structure.

The information in Figure 2 is already enough to justify use of a soft rather than stiff actuator. However, to make this justification more concrete, servo control loops are closed around each of the actuators, and the input responses and disturbance rejection are calculated. The derivations of the appropriate transfer functions are in Appendix B.

Figure 3 shows the closed-loop magnitude response for the stiff and soft actuators with their respective controllers. The responses show the usual features of a magnitude of 1 at low frequencies with a decreasing magnitude after some point. The soft actuator shows a bandwidth of more than 10 Hz. The stiff actuator shows a usable bandwidth of about 4 Hz[1], reflecting the limiting effect of the resonance near 10 Hz. The prominent 10 Hz resonance also indicates that there should be noticeable oscillations in the transient response.

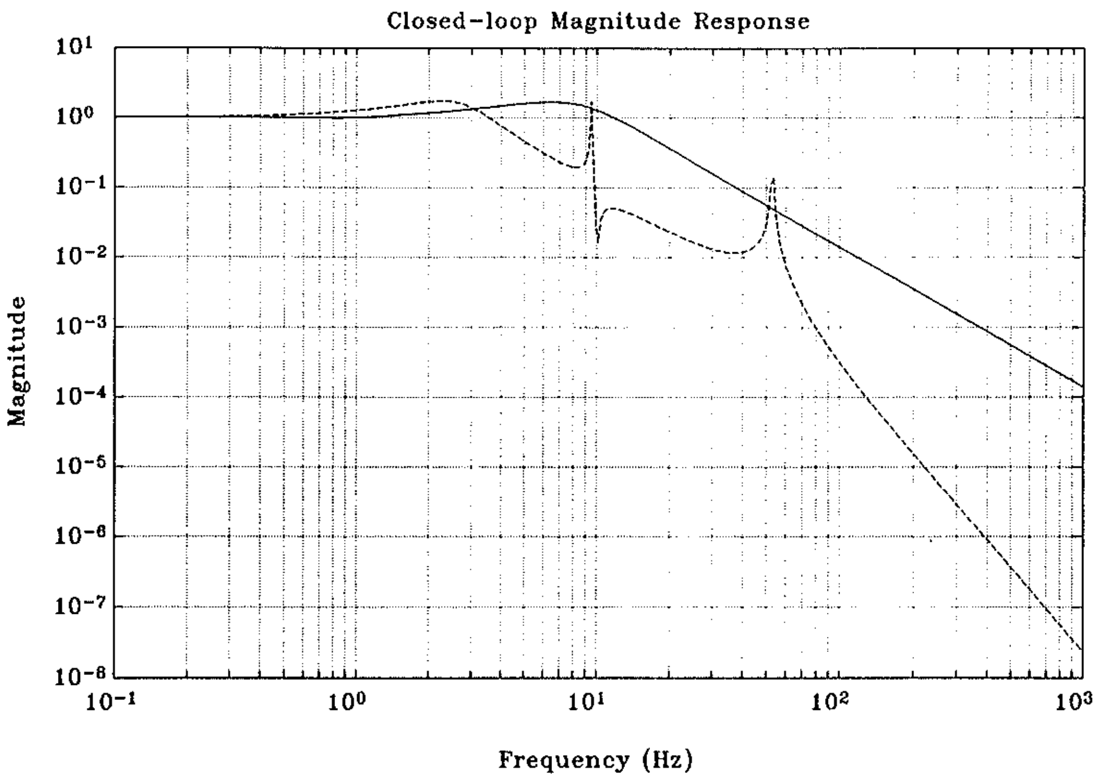


Figure 3.
Closed-loop magnitude responses for the soft (solid line) and stiff (dashed line) actuator cases. Note that the 3 dB bandwidth (4 Hz) of the stiff actuator case is considerably less than that of the soft actuator case (13 Hz).

[1]. Closed-loop bandwidth is distinct from the open-loop crossover frequency, the former being a rough measure of input response and the latter being a rather good indication of disturbance rejection as well as input response.

Figure 4 shows the step responses corresponding to the magnitude responses of Figure 3. The stiff actuator has a slower rise and a longer settling time than the soft actuator due mostly to its lower crossover frequency. The stiff actuator also shows some 10 Hz oscillation in the output. Both actuators show about 40% overshoot, but this is a minor concern. The overshoot can be eliminated by a prefilter or with a more sophisticated servo controller.

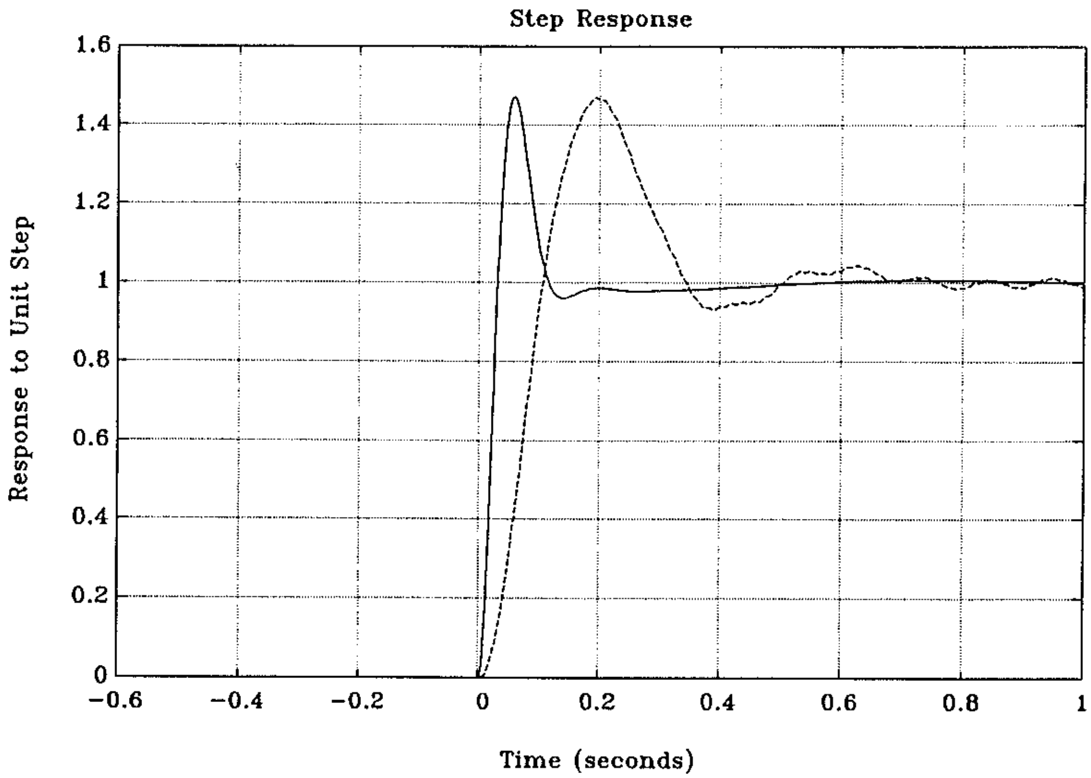


Figure 4.
Responses to a step input. Note that the soft actuator case (solid line) rises and settles faster than the stiff actuator case (dashed line). The considerable overshoot in both cases is a minor concern. It can be eliminated with a prefilter or with a more sophisticated controller than was used in this report.

The disturbance transmission magnitude for the two closed-loop systems is plotted in Figure 5. In this figure, the transmission is defined as the ratio of the amplitude of motion at m_1 to the force applied at m_2. This models the effect of the wind disturbing the secondary mirror support. As seen in the plot, disturbance transmission for the stiff actuator is greater (has worse disturbance rejection) than the soft actuator at all frequencies. The reasons for this result are the soft actuator's inherently good rejection of disturbances above about 1 Hz and its wider control bandwidth.

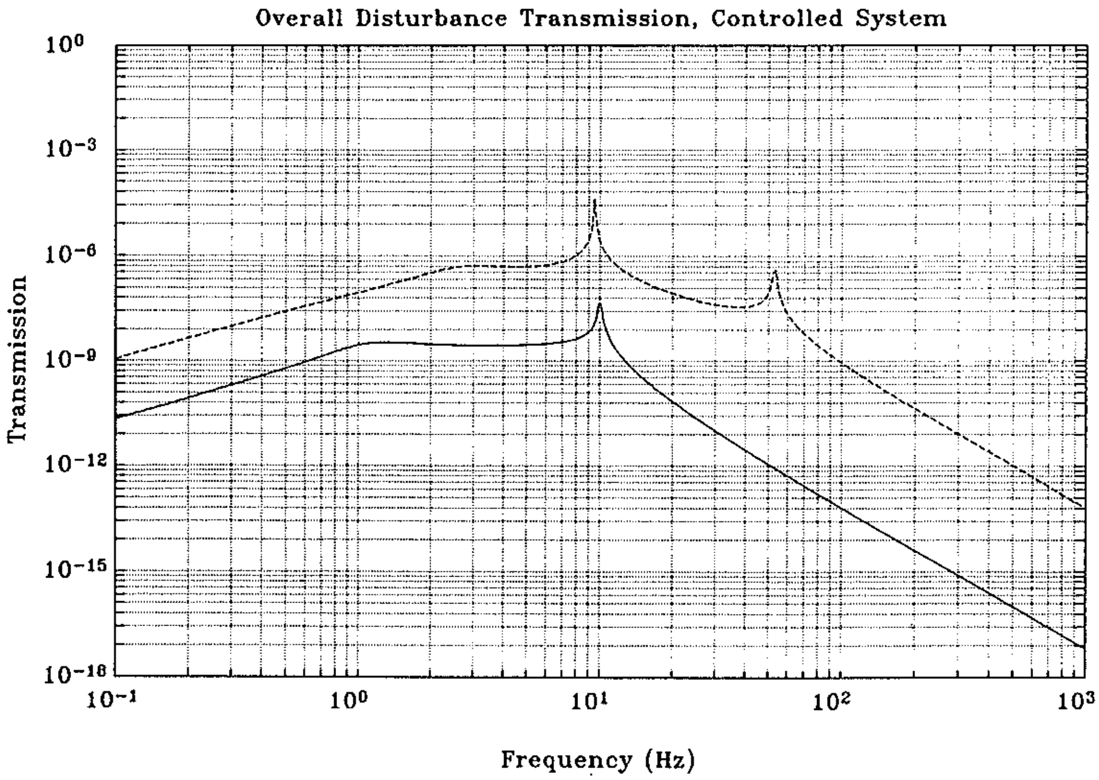


Figure 5.
Disturbance transmission in the frequency domain. This plot shows the ratio of the amplitude of motion at x_1 to the force applied at m_2. This models the secondary mirror motion due to wind blowing on the support structure. Note that the transmission for the soft actuator (solid line) is orders of magnitude less than that for the stiff actuator (dashed line).

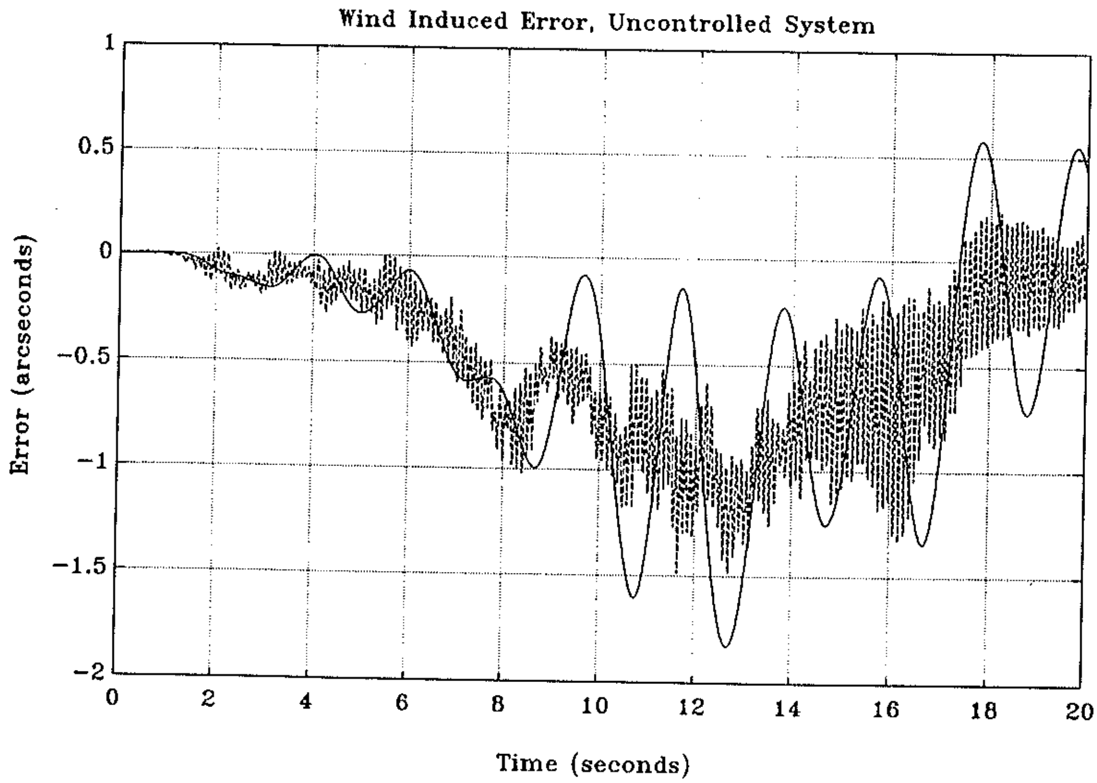
Figures 6a and 6b illustrate the disturbance rejection in the time domain. For these plots, the systems were subjected to a wind disturbance with a 1/f velocity spectrum (or equivalently, a $1/f^2$ force spectrum). Figure 6a shows the results for the uncontrolled (open-loop) system with the wind disturbance scaled to result in about 1.5 arcseconds of error[2]. Figure 6b show the results for the controlled (closed-loop) systems subjected to the same disturbance.

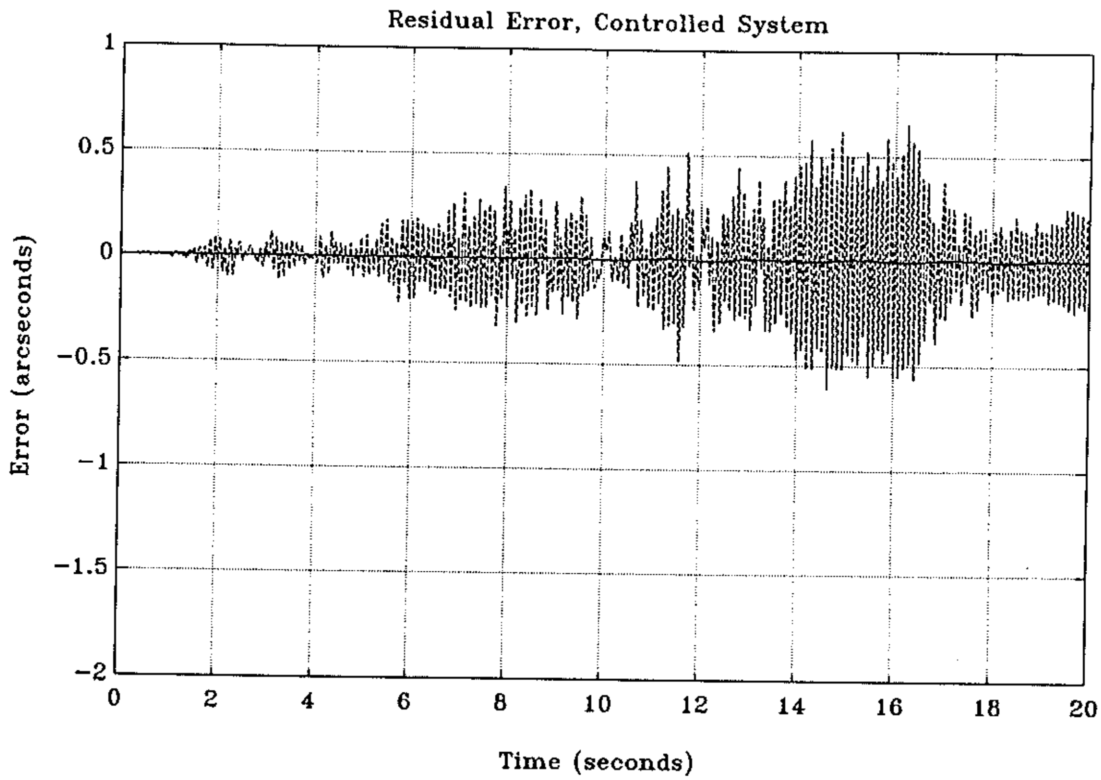
For the soft actuator, the errors are effectively reduced to zero. For the stiff actuator, considerable high-frequency disturbance remains.

There are several effects at work here. In the uncontrolled case of Figure 6a, the stiff actuator transmits much more motion at 10 Hz and the soft actuator transmits disturbances most noticeably at 1/2 Hz. Below 1/2 Hz, the two systems are subject to the same errors.

---

[2]. The scaling is really somewhat arbitrary since we are only interested here in the ratio of the initial to residual error.





Figures 6a,b.
Wind disturbance transmission in the time domain. With the loop open in Figure 6a, the transmitted errors for the soft actuator case (solid line) exceed those for the stiff actuator case (dashed line). The motion for the soft actuator is primarily at about 0.5 Hz; for the stiff actuator, about 10 Hz. For lower frequencies, the motions for the two cases are similar.

In Figure 6b with the loop closed, the low frequency motion for the stiff actuator case has been effectively removed, but the 10 Hz motion largely remains. For the soft actuator case, closing the loop effectively reduces the error to zero.

With the loop closed in Figure 6b, the stiff actuator still transmits at 10 Hz since its servo is ineffective at that frequency. The soft actuator, however, can strongly attenuate the 1/2 Hz disturbance. The 10 Hz oscillations are not a problem for the soft actuator since they are not transmitted in the first place.

DISCUSSION

Notably, it is the resonant mode of the secondary support structure that proved troublesome (in this case at 10 Hz), and not the resonance of the mirror on the actuator (at 1/2 Hz and about 50 Hz for the soft and stiff actuators, respectively). This illustrates that it is not just the frequency of the resonant modes that is important, but also how they enter the system dynamics (the mode shapes).

Given the results of this analysis, we might simply strive for an infinitely stiff support and infinitely soft actuator. Unsurprisingly, this approach is too simple and we would soon reach a point of diminishing returns in servo performance for two reasons. First, the filtering effect of the system inertia causes the wind induced position disturbances to fall off as $1/f^4$ (neglecting resonances), and they quickly become insignificant at the higher frequencies. This makes wideband servo control of lesser value. Second, we would find that the lowest bending mode of the secondary mirror is another servo bandwidth-limiting frequency. This bending frequency would be difficult to closely approach regardless of the rest of the structure.

In a practical case then, a servo crossover frequency of 8 Hz is most likely sufficient for disturbance rejection. It is also a reasonable goal in view of the anticipated telescope dynamics. This crossover frequency could be achieved with a 5 to 10 Hz support, a soft actuator, and a 25 to 30 Hz secondary mirror. If a wider bandwidth is needed for fast guiding, a correspondingly stiffer secondary mirror would be required.

Not addressed in this report is the feedback sensor for the systems under consideration. A guide camera may not produce measurements fast enough to allow an 8 Hz crossover, since a suitably bright guide star may not be near the object of interest. However, the combination of a guide camera with an accelerometer on the secondary mirror (and perhaps also on the telescope center section) would in fact provide the necessary information. These sensors would allow the secondary to be slaved to the primary mirror to reject disturbances in addition to steering the field of view to correct tracking errors.

CONCLUSION

For the control of a secondary mirror, a soft actuator is preferred over a stiff actuator. Since use of a stiff actuator results in the vibrational modes of the supporting structure limiting the servo bandwidth, a stiff actuator would be suitable only if the support structure mode frequencies began at 25 to 30 Hz.

A NOTE ON A PRACTICAL ACTUATOR DESIGN

Figure 7 shows a conceptual design for a secondary mirror actuator. It is based on a voice coil and a flexured linkage. Such an actuator could be designed to comfortably provide 0.5 mm of motion and 125 N of force at the output. Fatigue of the flexure is not a problem at these levels. The actuator concept assumes that the static weight of the mirror is carried by a vacuum in the cell or by some other counter-balancing system.

The design includes a load cell in the output to measure the applied force. Strictly speaking, this may not be necessary depending on the linearity of the voice coil. It may nonetheless be desirable for monitoring the forces on the support points.

The design also includes a position sensor capable of resolving 0.01 um (0.002 um at the output). This is needed for maintaining the low-frequency position of the mirror in the cell. It can also be used with the load cell in a compound feedback arrangement to tailor the output impedance of the actuator to the dynamics of the controlled system--an especially useful technique when the utmost in performance is desired.

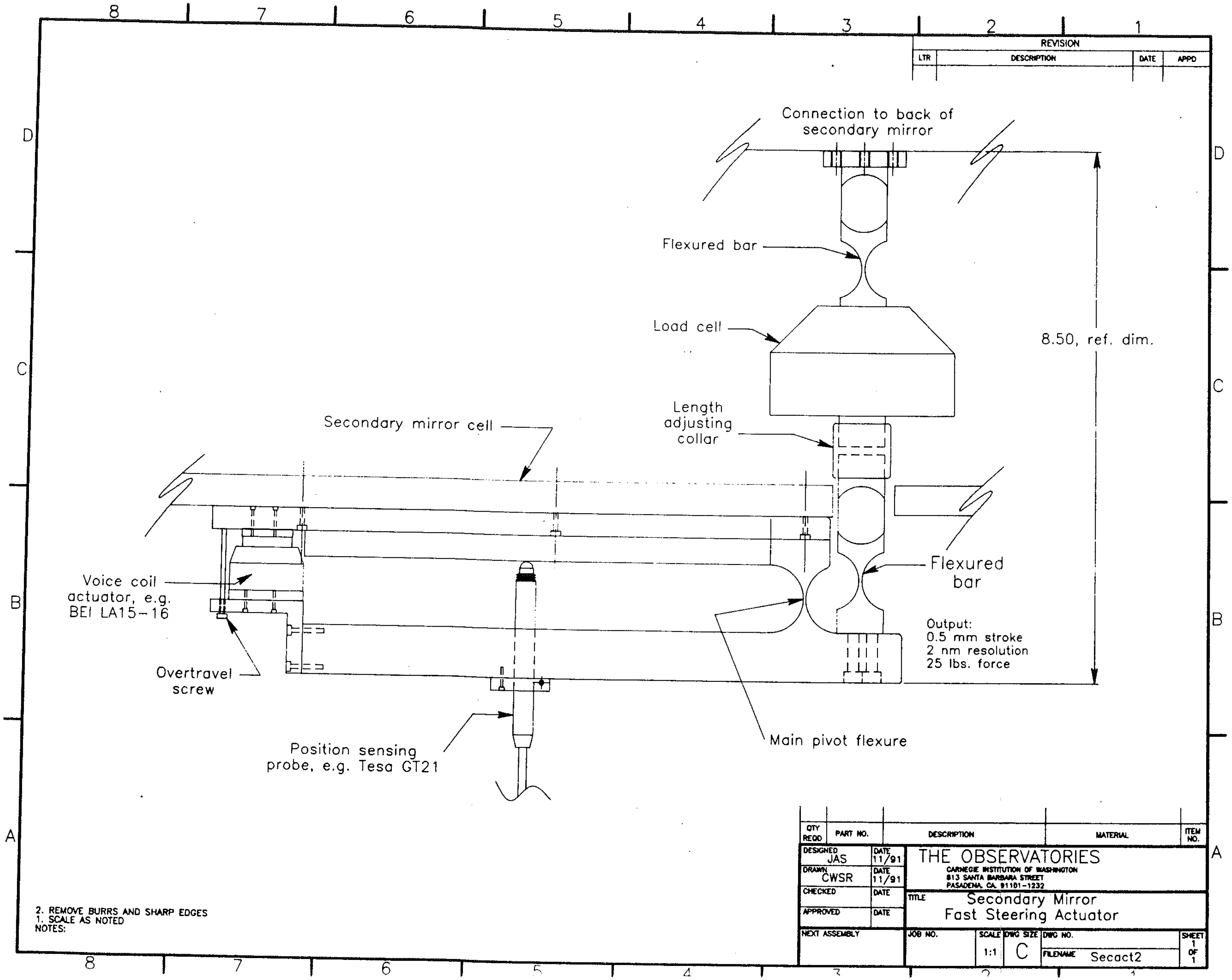
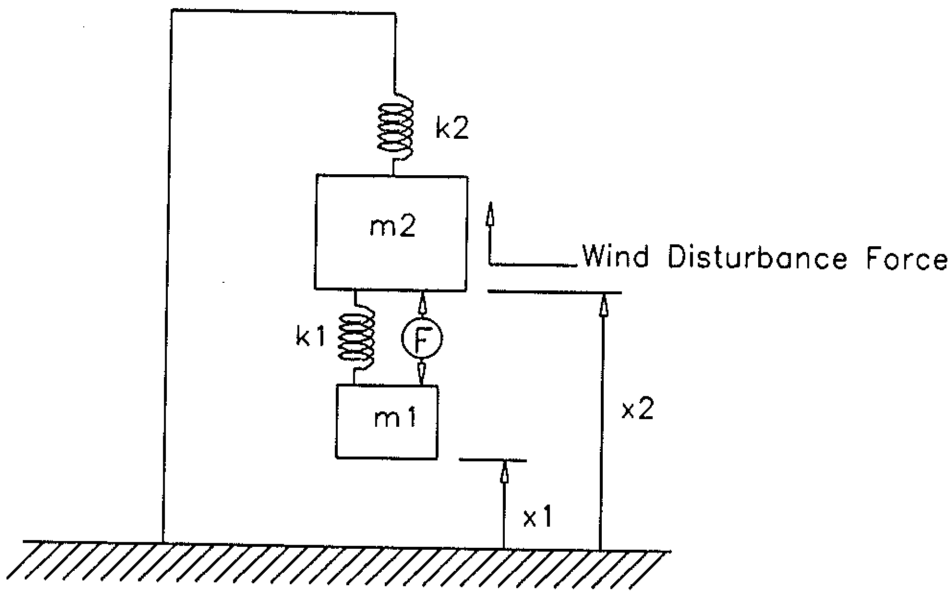

REVISION
LTR
DESCRIPTION
DATE
APPD
Connection to back of secondary mirror
Flexured bar
Load cell
8.50, ref. dim.
Length adjusting collar
Secondary mirror cell
Flexured bar
Voice coil actuator, e.g. BEI LA15-16
Output:
0.5 mm stroke
2 nm resolution
25 lbs. force
Overtravel screw
Main pivot flexure
Position sensing probe, e.g. Tesa GT21
QTY REQD
PART NO.
DESCRIPTION
MATERIAL
ITEM NO.
DESIGNED JAS
DATE 11/91
DRAWN CWSR
DATE 11/91
CHECKED
DATE
APPROVED
DATE
THE OBSERVATORIES
CARNEGIE INSTITUTION OF WASHINGTON
813 SANTA BARBARA STREET
PASADENA, CA 91101-1232
TITLE Secondary Mirror Fast Steering Actuator
NEXT ASSEMBLY
JOB NO.
SCALE 1:1
DWG SIZE C
DWG NO.
FILENAME Secact2
SHEET 1 OF 1
2. REMOVE BURRS AND SHARP EDGES
1. SCALE AS NOTED
NOTES:

Figure 7.

APPENDIX A

PLANT TRANSFER FUNCTION



Equations of Motion:

$$F+k_1(x_2-x_1)-m_1\ddot{x}_1=0 \quad (A.1)$$

$$-F+k_1(x_1-x_2)-k_2x_2-m_2\ddot{x}_2=0 \quad (A.2)$$

Taking the Laplace tranforms of (A.1) amd (A.2) and combining yields the transfer function

$$H=\frac{X_1}{F}=\frac{k_2+m_2s^2}{k_1k_2+[k_1m_2+(k_1+k_2)m_1]s^2+m_1m_2s^4} \quad (A.3)$$

(A.3) is the plant transfer function for the purposes of controller design. k_1 represents the actuator stiffness and is varied accordingly. For the models in this report, the parameters are:

m1 = 100 kg

m2 = 1000 kg

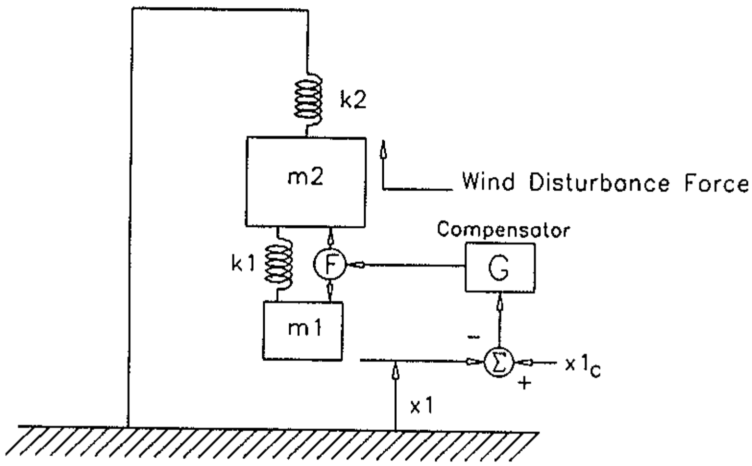
$k_2 = 3.97 \times 10^6 \frac{N}{M}$

k_1 soft $= 1000 \frac{N}{M}$

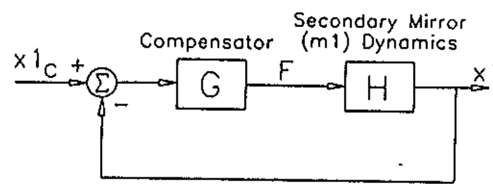
k_1 stiff $= 10^7 \frac{N}{M}$

APPENDIX B

COMPENSATORS, WIND INDUCED DISTURBANCES, DISTURBANCE TRANSMISSION, AND INPUT RESPONSE



Simplified Model



Block Diagram

Compensators

Compensator transfer function for soft actuator:

$$G_v = \frac{(2\pi)^3(s+3)(s+0.4\pm j)}{(1.8\times10^{-6})(s^3+32\pi s^2)} \qquad (B.1)$$

Compensator transfer function for stiff actuator:

$$G_p = \frac{(s+1.5\pi)}{(2.8\times10^{-10})(s^3+6\pi s^2)} \qquad (B.2)$$

The compensators above are used with the appropriate plant transfer function (from Appendix A) to determine the response of the system to various inputs.

Wind Induced Disturbances

For a disturbing wind force, W, introduced at m_2, the equations of motion are

$$k_1(x_2-x_1)-m_1\ddot{x}_1=0 \qquad (B.3)$$

$$W+k_1(x_1-x_2)-k_2x_2-m_2\ddot{x}_2=0 \quad (B.4)$$

Taking the Laplace transform of (B.3) and (B.4) and combining yields the transfer function

$$\frac{X_{1_{ol}}}{W}=\frac{k_1}{s^4m_1m_2+s^2(m_1(k_1+k_2)+k_1m_2)+k_1k_2} \qquad (B.5)$$

The magnitude of this expression gives the ratio of the amplitude of motion at x_1 to the wind force W as a function of the complex frequency, s, under the condition that the loop is open.

Disturbance Transmission

Referring to the block diagram at the begining of this appendix and using block diagram algebra, we find that the attenuation (closed loop vs. open loop) of a disturbance introduced at x_2 is

$$\frac{X_{1_{cl}}}{X_{1_{ol}}}=\frac{1}{1+GH} \qquad (B.6)$$

Where G is given by (B.1) or (B.2) depending on the actuator stiffness, and H is given by (A.3).

To find the residual closed-loop wind disturbance at x_1, we can multiply (B.5) by (B.6) to arive at

$$\frac{X_{1_{cl}}}{W}=\frac{k_1}{[s^4m_1m_2+s^2(m_1(k_1+k_2)+k_1m_2)+k_1k_2][1+GH]} \qquad (B.7)$$

The magnitude of this expression gives the ratio of the amplitude of motion at x_1 to the wind force, W, as a function of the complex frequency, s, under the condition that the loop is closed.

Input Response

Referring to the block diagram at the begining of this appendix, and using block diagram algebra, we find the input-output transfer function

$$\frac{X_1}{X_{1_c}}=\frac{GH}{1+GH} \qquad (B.8)$$

APPENDIX C

```
%Calculates the dynamics of a closed-loop secondary actuator system
%with actuators of varying stiffnesses.


if ~(exist('m1'))
        s=[' ',7];
        %Plant parameters.
        m1=100;
        m2=1000;
        k2=(2*pi*10)^2*m2;
        ksft=1000;
        kstf=1e7; %Actual piezo stiffness = 11e6 N/m.
        f=logspace(-1,3,200);

        %Calculate plant transfer function for soft spring.
        pnumsft=[m2 0 k2];
        pnrootssft=roots(pnumsft);
        pnrootssft=pnrootssft-0.0025*abs(pnrootssft); %Add a small amount of dam
        pnumsft=poly(pnrootssft);
        kpnumsft=k2/pnumsft(3);
        pnumsft=kpnumsft*pnumsft;

        pdensft=[m1*m2  0  (ksft*(m1+m2))+(k2*m1)  0  ksft*k2];
        pdrootssft=roots(pdensft);
        pdrootssft=pdrootssft-0.0025*abs(pdrootssft);
        pdensft=poly(pdrootssft);
        kpdensft=ksft*k2/pdensft(5);
        pdensft=kpdensft*pdensft;

        %Calculate plant transfer function for stiff actuator.
        pnumstf=[m2 0 k2];
        pnrootsstf=roots(pnumstf);
        pnrootsstf=pnrootsstf-0.0025*abs(pnrootsstf); %Add a small amount of dam
        pnumstf=poly(pnrootsstf);
        kpnumstf=k2/pnumstf(3);
        pnumstf=kpnumstf*pnumstf;

        pdenstf=[m1*m2  0  (kstf*(m1+m2))+(k2*m1)  0  kstf*k2];
        pdrootsstf=roots(pdenstf);
        pdrootsstf=pdrootsstf-0.0025*abs(pdrootsstf);
        pdenstf=poly(pdrootsstf);
        kpdenstf=kstf*k2/pdenstf(5);
        pdenstf=kpdenstf*pdenstf;

        %Calculate and plot plant transfer functions.
        [pmagsft,pphasesft]=bode(pnumsft,pdensft,2*pi*f);
        [pmagstf,pphasestf]=bode(pnumstf,pdenstf,2*pi*f);
        nstf=1/pmagstf(1);
        nsft=1/pmagsft(1);

        %Calculate compensator and open-loop transfer functions.
        cnumsft=poly(2*pi*[-3; -.4+j*1; -.4-j*1]);
        cnumsft=(1/1.8e-6)*cnumsft;
        cdensft=[1 2*pi*16 0 0];
        olnumsft=conv(cnumsft,pnumsft);
        oldensft=conv(cdensft,pdensft);

        cnumstf=[1 2*pi*.75];
        cnumstf=(1/2.8e-10)*cnumstf;
        cdenstf=[1 2*pi*3 0 0];
        olnumstf=conv(cnumstf,pnumstf);
        oldenstf=conv(cdenstf,pdenstf);
```

```
[olmagstf,olphasestf]=bode(olnumstf,oldenstf,2*pi*f);

%Calculate closed-loop transfer functions.
cldensft=[0 0 olnumsft]+oldensft;
clnumsft=olnumsft;
[clmagsft,clphasesft]=bode(clnumsft,cldensft,2*pi*f);
cldenstf=[0 0 0 0 olnumstf]+oldenstf;
clnumstf=olnumstf;
[clmagstf,clphasestf]=bode(clnumstf,cldenstf,2*pi*f);

%Calculate wind transmission to m1.  System is uncontrolled.
dtnumsft=ksft;
dtdensft=pdensft;
[dtmagsft,dtphasesft]=bode(dtnumsft,dtdensft,2*pi*f);
dtnumstf=kstf;
dtdenstf=pdenstf;
[dtmagstf,dtphasestf]=bode(dtnumstf,dtdenstf,2*pi*f);


%Calculate disturbance rejection transfer functions.
%This is really the transmission of wind induced position errors
%to m1 with the system uncontrolled.
drnumsft=conv(pdensft,cdensft);
drdensft=conv(pdensft,cdensft)+[0 0 conv(cnumsft,pnumsft)];
[drmagsft,drphasesft]=bode(drnumsft,drdensft,2*pi*f);
drnumstf=conv(pdenstf,cdenstf);
drdenstf=conv(pdenstf,cdenstf)+[0 0 0 0 conv(cnumstf,pnumstf)];
[drmagstf,drphasestf]=bode(drnumstf,drdenstf,2*pi*f);

%Calculate the overall transfer function from the wind to the
%error in the position of m1 with the system controlled.
wtnumsft=conv(dtnumsft,drnumsft);
wtdensft=conv(dtdensft,drdensft);
[wtmagsft,wtphasesft]=bode(wtnumsft,wtdensft,2*pi*f);
wtnumstf=conv(dtnumstf,drnumstf);
wtdenstf=conv(dtdenstf,drdenstf);
[wtmagstf,wtphasestf]=bode(wtnumstf,wtdenstf,2*pi*f);

%Form the wind spectrum.
%yfm=1 ./ f.^2; %Forms 1/f.^2 force spectrum.
%yfa=2*pi*zeros(yfm); %Form a phase function.
%yf=yfm.*exp(j*yfa); %Wind force spectrum (complex).
wnum=[(2*pi)^2]; %Normalized to 1 at 1 Hz.
wden=[1 0 0];
[yfm,yfa]=bode(wnum,wden,2*pi*f);


%Convolve the wind with the system dynamics. Uncontrolled case.
ucwerrnumstf=conv(dtnumstf,wnum);
ucwerrdenstf=conv(dtdenstf,wden);
[ucwerrmagstf,ucwerrargstf]=bode(ucwerrnumstf,ucwerrdenstf,2*pi*f);
ucwerrnumsft=conv(dtnumsft,wnum);
ucwerrdensft=conv(dtdensft,wden);
[ucwerrmagsft,ucwerrargsft]=bode(ucwerrnumsft,ucwerrdensft,2*pi*f);

%yucwerrstf=(polyval(dtnumstf,2*pi*j*f)./polyval(dtdenstf,2*pi*j*f)).*yf
%ucwerrmagstf=abs(yucwerrstf);
%ucwerrmagstf=ucwerrmagstf*(1.5/max(ucwerrmagstf)); %Normalize to 1.5 ar
%yucwerrsft=(polyval(dtnumsft,2*pi*j*f)./polyval(dtdensft,2*pi*j*f)).*yf
%ucwerrmagsft=abs(yucwerrsft);
%ucwerrmagsft=ucwerrmagsft*(1.5/max(ucwerrmagstf));
%yf=yf*(1.5/max(ucwerrmagstf));

%Convolve the wind with the system dynamics. Controlled case

[werrmagstf,werrargstf]=bode(werrnumstf,werrdenstf,2*pi*f);
werrnumsft=conv(wnum,wtnumsft);
werrdensft=conv(wden,wtdensft);
[werrmagsft,werrargsft]=bode(werrnumsft,werrdensft,2*pi*f);
%ywerrstf=(polyval(wtnumstf,2*pi*j*f)./polyval(wtdenstf,2*pi*j*f)).*yf;
%werrmagstf=abs(ywerrstf);
%ywerrsft=(polyval(wtnumsft,2*pi*j*f)./polyval(wtdensft,2*pi*j*f)).*yf;
%werrmagsft=abs(ywerrsft);
```

```
        %Calcualte a "wind".
        t=0:.005:20;
        t1=t(1:1000);
        t2=t(1000:2000);
        t3=t(2000:3000);
        t4=t(3000:4001);
        t=[t1 t2(2:length(t2)) t3(2:length(t3)) t4(2:length(t4))];
        rand('normal');
        wraw=rand(length(t),1); %Generate "broadband" wind.
        wnum=1;
        wden=[1 0];
        wfilt=lsim(wnum,wden,wraw,t); %Filter random sequence for 1/f wind veloc
        wfilt=wfilt/max(abs(wfilt)); %Extremum normalized to 1.
        w=real((wfilt.^2).*sign(wfilt)); %Wind forces are proportional to veloci
        w1=w(1:1000);
        w2=w(1000:2000);
        w3=w(2000:3000);
        w4=w(3000:4001);

        %Calculate uncompensated wind errors (no control).
        [adtsft,bdtsft,cdtsft,ddtsft]=tf2ss(dtnumsft,dtdensft);
        [adtstf,bdtstf,cdtstf,ddtstf]=tf2ss(dtnumstf,dtdenstf);
        [ucwerrsft1,xdtsft1]=lsim(adtsft,bdtsft,cdtsft,ddtsft,w1,t1);
        [ucwerrsft2,xdtsft2]=lsim(adtsft,bdtsft,cdtsft,ddtsft,w2,t2,xdtsft1(leng
        [ucwerrsft3,xdtsft3]=lsim(adtsft,bdtsft,cdtsft,ddtsft,w3,t3,xdtsft2(leng
        [ucwerrsft4,xdtsft4]=lsim(adtsft,bdtsft,cdtsft,ddtsft,w4,t4,xdtsft3(leng
        [ucwerrstf1,xdtstf1]=lsim(adtstf,bdtstf,cdtstf,ddtstf,w1,t1);
        [ucwerrstf2,xdtstf2]=lsim(adtstf,bdtstf,cdtstf,ddtstf,w2,t2,xdtstf1(leng
        [ucwerrstf3,xdtstf3]=lsim(adtstf,bdtstf,cdtstf,ddtstf,w3,t3,xdtstf2(leng
        [ucwerrstf4,xdtstf4]=lsim(adtstf,bdtstf,cdtstf,ddtstf,w4,t4,xdtstf3(leng

        %Normalize the wind induced errors.
        ucwerrstf=[ucwerrstf1; ucwerrstf2(2:length(t2)); ucwerrstf3(2:length(t3)
        ucwerrsft1=real(1.5*ucwerrsft1/max(abs(ucwerrstf)));
        ucwerrsft2=real(1.5*ucwerrsft2/max(abs(ucwerrstf)));
        ucwerrsft3=real(1.5*ucwerrsft3/max(abs(ucwerrstf)));
        ucwerrsft4=real(1.5*ucwerrsft4/max(abs(ucwerrstf)));
        ucwerrstf1=real(1.5*ucwerrstf1/max(abs(ucwerrstf)));
        ucwerrstf2=real(1.5*ucwerrstf2/max(abs(ucwerrstf)));
        ucwerrstf3=real(1.5*ucwerrstf3/max(abs(ucwerrstf)));
        ucwerrstf4=real(1.5*ucwerrstf4/max(abs(ucwerrstf)));
        ucwerrsft=[ucwerrsft1; ucwerrsft2(2:length(t2)); ucwerrsft3(2:length(t3)
        ucwerrstf=[ucwerrstf1; ucwerrstf2(2:length(t2)); ucwerrstf3(2:length(t3)

        %Attenuate the errors by the disturbance rejection transfer function.
        [adrsft,bdrsft,cdrsft,ddrsft]=tf2ss(drnumsft,drdensft);
        [adrstf,bdrstf,cdrstf,ddrstf]=tf2ss(drnumstf,drdenstf);
        [werrsft1,xdrsft1]=lsim(adrsft,bdrsft,cdrsft,ddrsft,ucwerrsft1,t1);
        [werrsft2,xdrsft2]=lsim(adrsft,bdrsft,cdrsft,ddrsft,ucwerrsft2,t2,xdrsft
        [werrsft3,xdrsft3]=lsim(adrsft,bdrsft,cdrsft,ddrsft,ucwerrsft3,t3,xdrsft
        [werrsft4,xdrsft4]=lsim(adrsft,bdrsft,cdrsft,ddrsft,ucwerrsft4,t4,xdrsft
        [werrstf1,xdrstf1]=lsim(adrstf,bdrstf,cdrstf,ddrstf,ucwerrstf1,t1);
        [werrstf2,xdrstf2]=lsim(adrstf,bdrstf,cdrstf,ddrstf,ucwerrstf2,t2,xdrstf
        [werrstf3,xdrstf3]=lsim(adrstf,bdrstf,cdrstf,ddrstf,ucwerrstf3,t3,xdrstf


        clear xdrstf4 xdrstf3 xdrstf2 xdrstf1 xdrsft4 xdrsft3 xdrsft2 xdrsft1
        clear xdtstf4 xdtstf3 xdtstf2 xdtstf1 xdtsft4 xdtsft3 xdtsft2 xdtsft1

        werrsft=[werrsft1; werrsft2(2:length(t2)); werrsft3(2:length(t3)); werrs
        werrstf=[werrstf1; werrstf2(2:length(t2)); werrstf3(2:length(t3)); werrs

end

pack

loglog(f,pmagsft*nsft,f,pmagstf*nstf,'--'),grid
title('Plant Magnitude Responses (DC Normalized to 1)')
xlabel('Frequency (Hz)')
ylabel('Normalized Magnitude')
pause
```

```
A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot1
end

semilogx(f,olphasesft,f,olphasestf,'--'),grid
title('Open-loop Phase Response')
xlabel('Frequency (Hz)')
ylabel('Phase (degrees)')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot2
end

axis([-1 2 -5 5]);
loglog(f,olmagsft,f,olmagstf,'--'),grid
title('Open-loop Magnitude Response')
xlabel('Frequecy (Hz)')
ylabel('Magnitude')
axis; %Axes free-ranged.
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot3
end

loglog(f,clmagsft,f,clmagstf,'--'),grid;
title('Closed-loop Magnitude Response')
xlabel('Frequency (Hz)')
ylabel('Magnitude')
pause

        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot4
end

semilogx(f,clphasesft,f,clphasestf,'--'),grid
title('Closed-loop Phase Response')
xlabel('Frequency (Hz)')
ylabel('Phase (degrees)')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot5
end

loglog(f,yfm),grid
title('Wind Force Spectrum')
xlabel('Frequency (Hz)')
ylabel('Normalized Amplitude')
pause
```

```
A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot6
end

loglog(f,dtmagsft,f,dtmagstf,'--'),grid
title('Wind Disturbance Transmission, Uncontrolled System')
xlabel('Frequency (Hz)')
ylabel('Transmission')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot7
end

loglog(f,ucwerrmagsft,f,ucwerrmagstf,'--'),grid
title('Uncorrected Wind Error Spectrum')
xlabel('Frequency (Hz)')
ylabel('Transmission')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'


loglog(f,drmagsft,f,drmagstf,'--'),grid
title('Position Disturbance Transmission, Controlled System')
xlabel('Frequency (Hz)')
ylabel('Transmission')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot9
end

loglog(f,werrmagsft,f,werrmagstf,'--'),grid
title('Residual Wind Error Spectrum')
xlabel('Frequency (Hz)')
ylabel('Transmission')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot10
end


loglog(f,wtmagsft,f,wtmagstf,'--'),grid
title('Overall Disturbance Transmission, Controlled System')
xlabel('Frequency (Hz)')
ylabel('Transmission')
pause
```

```
A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot11
end

plot(t,wfilt),grid
title('Wind Velocity vs. Time')
xlabel('Time (seconds)')
ylabel('Wind Velocity (arbitrary units)')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot12
end

indices=[1:2:length(t)]; %Used to decimate the large data.
plot(t(indices),ucwerrsft(indices),t(indices),ucwerrstf(indices),'--'), grid

ylabel('Error (arcseconds)')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot13
end

plot(t(indices),werrsft(indices),t(indices),werrstf(indices),'--'), grid
title('Residual Error, Controlled System')
xlabel('Time (seconds)')
ylabel('Error (arcseconds)')
pause

A=input('Save the plot? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        meta d:\matlab\secact\plot14
end
```