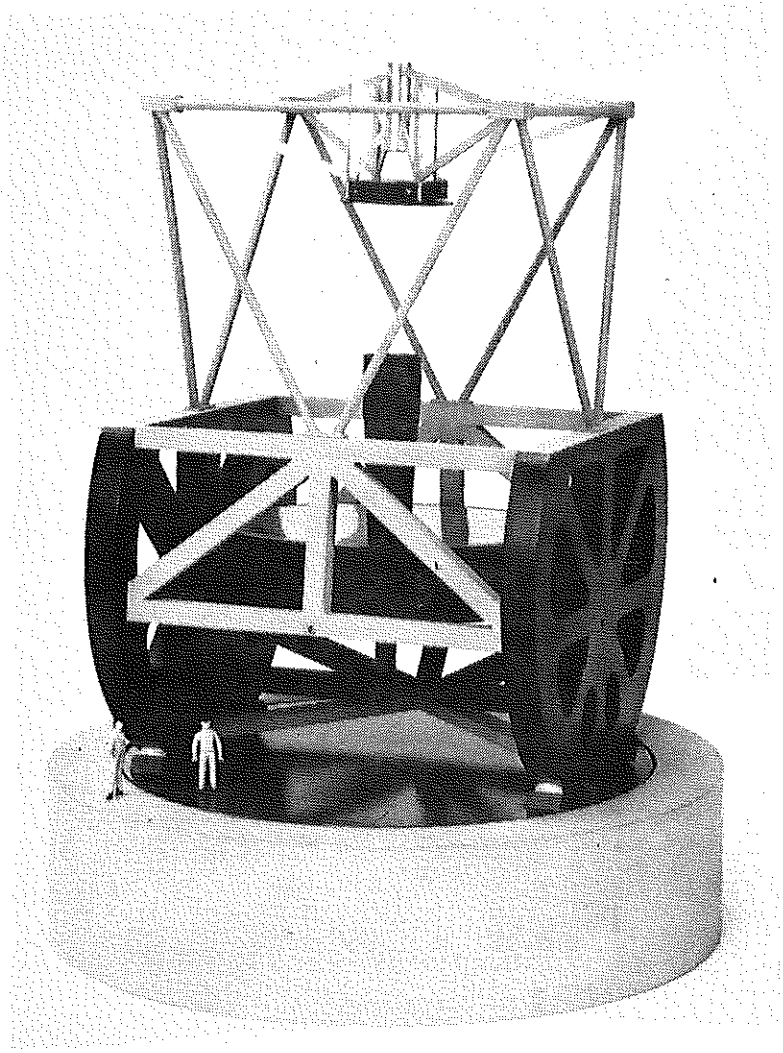
MAGELLAN PROJECT

University of Arizona | Carnegie Institution of Washington | The Johns Hopkins University



Torque Perturbations in the Magellan Main Drive Motors

J. Alan Schier
The Observatories of the
Carnegie Institution of Washington
Pasadena, California

March 20, 1990
No. 16

SUMMARY

An analysis has been performed to evaluate the effect of drive motor torque perturbations on telescope pointing performance. The analysis assumes values for telescope mass and dynamic properties and motor performance that are representative of the currently proposed Magellan design. The results indicate that the selected motors are suited to meeting the proposed tracking requirement of 0.05 arcsec. over 1 minute and 0.1 arcsec. for longer periods. The analysis also gives a measure of sensitivity of the pointing performance to the structural dynamics of the telescope; namely, if the frequency of the first important structural mode were to drop from the assumed 7.5 Hz to 5 Hz, the pointing performance requirement could not be met without some additional sophistication in the drive controls.

This analysis addresses only the disturbances due to motor cogging and ripple. Inasmuch as there are other disturbances which have not been considered, this analysis is not a complete assessment of the anticipated telescope pointing performance.

DESCRIPTION OF THE MODEL

The most significant elements in the model are the following:

1. Third order controller, including an integral term.
2. Open-loop crossover frequency equal to three tenths of that of the first important structural mode (e.g., 2.25 Hz for a 7.5 Hz first mode).
3. Telescope inertias:
   Azimuth.....4.5e6 kg*m^2.
   Altitude.....3.0e6 kg*m^2.
4. Drive ratio = 50:1.
5. Motor cogging torque = 0.74 Nm (1 ft*lb) peak to peak at 144 cycles per motor revolution.
6. Motor ripple torque = 20% of applied torque.
7. Two motors per axis.
8. Motor torque perturbations in phase.
9. Continuous time and position.

A block diagram of one axis of the system is shown in figure 1, and the Matlab file for generating the results presented here is included as an appendix.

ERRATUM

Page 1, item 5 under Desciption of the Model should read, "... = 0.74 Nm (.54 ft*lb) peak to peak...". The results of the analysis are unchanged.

The features of this model are all conservative with the exception of elements number 2 and 9 in the list above. Number 9, the assumption of continuous time and position, is nonconservative in that the effects of the digital discretization of both time and position will cause pointing errors that are not apparent in this analysis. However, it has been assumed that in the final system the sampling frequency will be made significantly faster (a factor of ten or more) than the crossover frequency, and the position resolution will be made fine enough so that position quantization errors will be tolerable. To the extent that these assumptions are correct, these errors will be of secondary importance.

Item number 2 in the list above, a crossover frequency of 0.3 times the first structural frequency, is achievable but probably not without a more sophisticated controller than is considered in this analysis. Aside from permitting a wider control bandwidth, the added complexity of such a controller would have only a secondary effect on the performance of the system, and this complexity was therefore not considered in this analysis. It must be noted however, that the chosen crossover frequency is a nonconservative estimate, and there is a significant chance--perhaps 15%-- that it cannot reasonably be achieved. If this were to happen, the performance will degrade as will be illustrated in the section which follows, and other augmentations to the system may be needed to insure the desired performance.

ANALYTICAL RESULTS

Figure 2 shows the pointing errors due to motor torque perturbations for a system with an open-loop crossover frequency of 2.25 Hz on an assumed 7.5 Hz structure. This data is for a 50 second period beginning 4000 seconds after passing within 0.5 degrees of the zenith. The oscillatory error in the altitude direction show a peak amplitude of about 0.012 arcseconds and is due almost entirely to the motor cogging torque. The azimuth error shows no oscillations since the motors do not complete a full cycle of the cogging torque in this 50 second period. The motor ripple torque has a negligible effect since it is generated as a fraction of the applied torque and the motor torques are very small in this case.

Figure 3 shows a similar plot of pointing errors for a system in which the open-loop crossover frequency has been lowered to 1.5 Hz. This could correspond to a 5 Hz structure or to a simple controller on a 7.5 Hz structure. The result is that the dominant error--the altitude axis error--has an amplitude of about 0.039 arcseconds. This is more than a factor of three worse than the previous case and would probably be considered unacceptable if left uncorrected.

Figure 4 shows a third similar plot in which the open-loop crossover frequency has been raised to 3 Hz, a value which should be achievable on a 10 Hz structure. The result is that the amplitude of the altitude axis error has been reduced to less than 0.006 arcseconds. This is more than a factor of 2 better than the results in figure 2 with a 2.25 Hz crossover frequency.

IMPLICATIONS OF THE ANALYTICAL RESULTS

These results clearly demonstrate the desirability of wideband control. In this case, reducing the crossover frequency by one third from 2.25 Hz to 1.5 Hz more than tripled the resulting error. This has clear implications for the dynamics of the telescope structure. Roughly speaking, if the frequency of the first important structural mode drops by 30%, the resulting pointing disturbance can rise by as much as a factor of three.

In the case with a 3 Hz crossover frequency, the error due to motor cogging was reduced to an almost insignificant level. Since cogging will not be the only disturbance in the system, the actual errors would be larger than shown in figure 3. Nonetheless, the wideband control will attentuate many of the additional disturbances in a manner similar to that demonstrated for the cogging torque.

Although not shown explicitly, these results also indicate that if disturbances such as are caused by cable wraps and rolling friction become excessive so that the motor is required to continuously exert an appreciable torque, the motor ripple torque will begin to introduce a significant error. Moreover, such disturbances are usually a particular problem when direction reversals are involved such as will occur in fighting errors due to the wind. The implication then is that parasitic disturbance torques are to be minimized and made repeatable within practical limits. Minimizing the disturbances has the obvious effect of minimizing the resulting errors, and if any remaining disturbances are repeatable, they can be corrected in an open loop fashion. This usually implies that hysteresis and backlash must be held to a bare minimum.

CONCLUSIONS

The results of this analysis indicate that the proposed drive motors are suited to meeting the Magellan pointing requirements. When combined with the rest of the control system, the effect of motor cogging and ripple on pointing are expected to be a significant but tolerable part of the overall pointing error budget. Nevertheless, it should be noted that the analysis indicates that there are areas in which changes could be made in order to improve performance and make the system more robust. These areas are noted in the next section.

SUGGESTIONS FOR FURTHER CONSIDERATION

In the interest of improving performance and making the system more tolerant of disturbances and parameter variations, some measures worth considering are the following:

1. Stiffen the structure and widen the control bandwidth. One of the best weapons for fighting wind disturbances is to start with a stiff structure. One of the best weapons for fighting disturbances in general--whether they originate from the wind, the motors, or elsewhere-- is control system bandwidth, and typically it is the structural dynamics which limit the available bandwidth. It is therefore recommended that the current 7.5 Hz first structural mode be considered a minimum acceptable frequency for the Magellan design and that consideration be given to the possibility of further raising that frequency.

2. Guide with the secondary mirror. It is expected that the control bandwidth of the secondary mirror can be made significantly higher than that of the main drives. This relatively wideband system can then be used to reject disturbances beyond the range where the main drives are effective in addition to further attenuating any residual low frequency errors.

3. Compensate for motor torque perturbations in an open-loop fashion. Since the motor torque perturbations are a largely predictable function of motor position and current, it is possible encode the motor position, measure the motor current, and use this information to inject a correction signal at the input of the motor drive amplifier. This can be simpler, cheaper, and more effective than paying the additional money for a specially engineered motor and amplifier.

4. Design a more sophisticated controller. This analysis considered a third order controller which may be viewed as a minimum usable configuration. With a modest increase in complexity, a more sophisticated controller could be expected to produce improved results, especially in the frequency range below 0.5 Hz. As an example, a reasonable target would be to improve the disturbance rejection at 0.1 Hz by a factor of four.

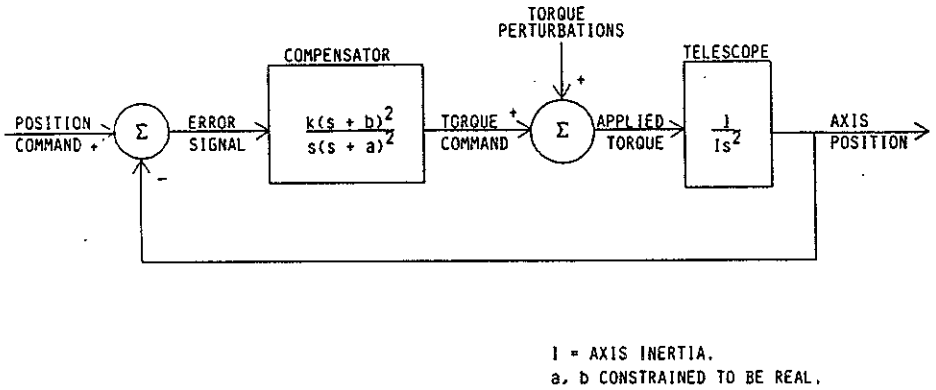


Figure 1 - Block Diagram

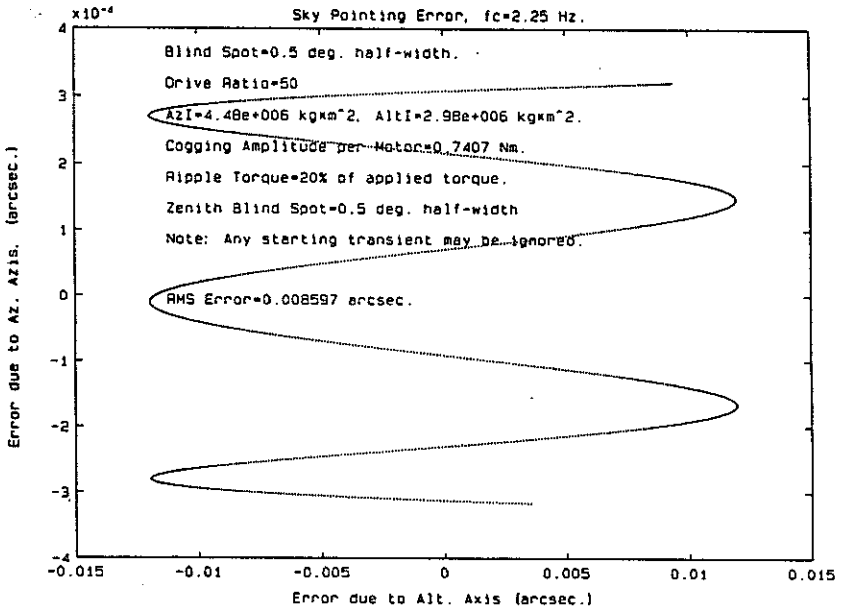


Figure 2 - Pointing Error, 2.25 Hz Case

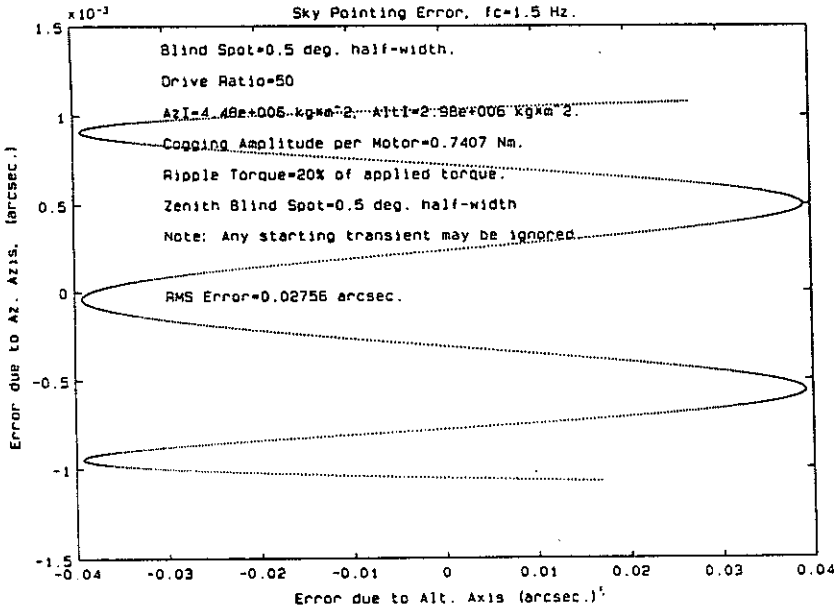


Figure 3 - Pointing Error, 1.5 Hz

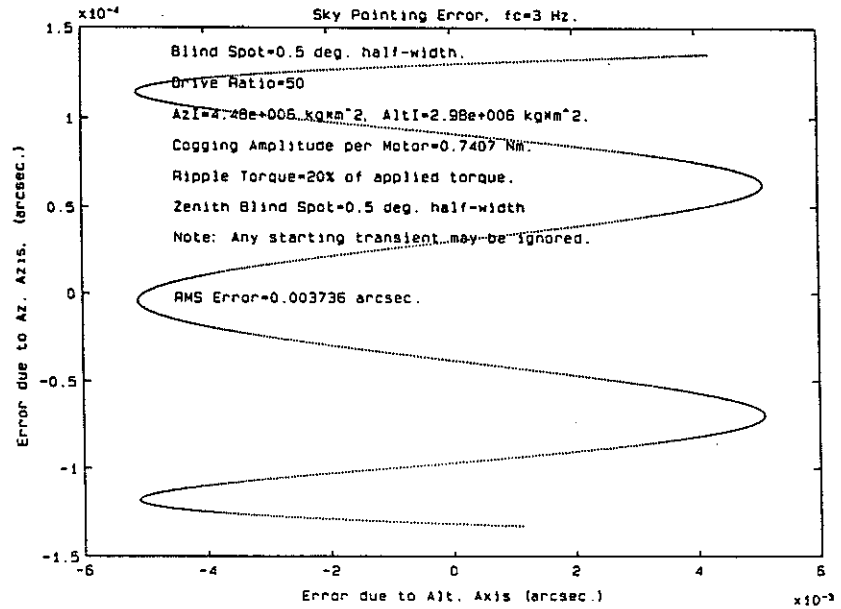


Figure 4 - Pointing Error, 3 Hz Case

APPENDIX

```
%  File: motdist.m (motdist.bak)
%  This file is used to figure the dynamic response of alt-az telescope
%  axes resulting from cogging and ripple torques at the drive motor.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The following section is parameter input from the keyboard.             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

AzI=input('Az. axis moment of inertia? [4.48e6 Nm]:');
if isempty(AzI)
        AzI=4.48e6;
end

AltI=input('Alt. axis moment of inertia? [2.98e6 Nm]:');
if isempty(AltI)
        AltI=2.98e6;
end

fc=input('Open loop crossover frequency? [1.5 Hz]:');
if isempty(fc)
        fc=1.5;
end

cone=input('Blind spot half-width? [0.5 deg.]:');
if isempty(cone)
        cone=0.5;
end

lat=input('Local latitude? [30 deg.]:');
if isempty(lat)
        lat=30;
end


tb=input('Beginning time? [0 sec.]:');
if isempty(tb)
        tb=0;
end

te=input('Ending time? [50 sec.]:');
if isempty(te)
        te=50;
end

ti=(te-tb)/2000;  %Calculate time increment.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The following section calculates the azimuth transfer functions.        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wc=2*pi*fc;                 %Crossover frequency.
Azk=(25*AzI*wc^2)/6;        %Compensator gain.
Aza=5*wc;                   %Compnesator pole position.
Azb=wc/5;                   %Compensator zero position.

Aznumoltf=[Azk 2*Azk*Azb Azk*Azb^2];    %Open loop numerator.
Azdenoltf=[AzI Aza*AzI 0 0 0];          %Open loop denominator.

Aznumdj=[1 Aza 0];          %Disturbance rejection numerator.
```

```
Azdendj=[AzI Aza*AzI Azk 2*Azk*Azb  Azk*Azb^2]; %Disturbance rej. denominator.

Aznumcltf=[Azk 2*Azk*Azb Azk*Azb^2];  %Closed loop numerator.
Azdencltf=[AzI Aza*AzI Azk 2*Azk*Azb Azk*Azb^2];  %Closed loop denominator.

t=(tb:ti:te)';  %Time vector.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The following section calculates az. dynamic response to motor dist.%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[altpos,altrate,altaccel]=altpra(lat,cone,t); % Alt. vlaues returned in radian m
[azpos,azrate,azaccel]=azpra(lat,cone,t); %Az. values returned in radian measure
Azta=AzI*azaccel;  %Computed motor torque applied to azimuth.
pr=.2;  %Fractional amout of ripple torque.
fcg=144; %Cogging frequency in cycles per revolution.
fr=144;  %Ripple frequency in cycles per revolution.
ca=1/1.35;  %Cogging amplitude at the motor(per motor), peak to peak value in Nm
N=50;  %Drive ratio.
Aztr=abs((pr)*Azta.*abs(sin((fr/2)*N*azpos)));  %Computed ripple torque.
Aztc=(ca)*N*(sin((fcg)*N*azpos));  %Computed cogging torque.
Aztcr=Aztc+Aztr;  %Total torque perturbations.
Azx=lsim(Aznumdj,Azdendj,Aztcr,t);  %Calculate az. axis response.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The following section plots the various results.                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


plot(t,Azx/4.85e-6),grid;
title(['Azimuth Position Error, fc=',num2str(fc),' Hz.'])
xlabel('Time Past Closest Approach to Zenith, (sec.)')
ylabel('Azimuth Position Error, (arcsec.)')
text(.2,.85,['Drive Ratio=',num2str(N)],'sc')
text(.2,.8,['AzI=',num2str(AzI),' kg*m^2.'],'sc')
text(.2,.75,['Cogging Amplitude per Motor=',num2str(ca),' Nm.'],'sc')
text(.2,.7,['Ripple Torque=',num2str(pr*100),'% of applied torque.'],'sc')
text(.2,.65,['Zenith Blind Spot=',num2str(cone),' deg. half-width'],'sc')
text(.2,.6,['Note: Any starting transient may be ignored.'],'sc')
text(.2,.5,['Standard Deviation=',num2str(std(Azx(500:1500)/4.85e-6)),' arcsec.'
pause


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

if A~='N'
        meta d:\matlab\mgn\drives\driveplot
end


A=input('Other plots? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
```

```
A=input('Sky error due to azimuth? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        azptgerr=cos(altpos).*(Azx/4.85e-6);
        plot(t,azptgerr),grid;
        title(['Sky Pointing Error due to Az., fc=',num2str(fc),' Hz.'])
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Pointing Error, (arcsec.)')
        text(.2,.85,['Drive Ratio=',num2str(N)],'sc')
        text(.2,.8,['AzI=',num2str(AzI),' kg*m^2.'],'sc')
        text(.2,.75,['Cogging Amplitude per Motor=',num2str(ca),' Nm.'],
        text(.2,.7,['Ripple Torque=',num2str(pr*100),'% of applied torqu
        text(.2,.65,['Zenith Blind Spot=',num2str(cone),' deg. half-widt
        text(.2,.6,['Note: Any starting transient may be ignored.'],'sc'
        text(.2,.5,['Standard Deviation=',num2str(std(azptgerr(500:1500)
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Disturbance torques? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,Aztr,t,Aztc),grid;
        title(['Disturbance Torque, Cogging Amp.=',num2str(ca),...
        ' Nm, Ripple=',num2str(pr*100),'%'])
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Disturbance Torque, (Nm)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Computed torque? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,Azta),grid;
        title(['Computed Torque, Azimuth Inertia=',num2str(AzI),' kg*m^2
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Computed Torque, (Nm)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
```

```
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Azimuth position? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,azpos*180/pi),grid;
        title(['Azimuth Position, Blind Spot Half-Width =',num2str(cone)
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Azimuth Position, (deg.)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end


A=input('Azimuth rate? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,azrate*180/pi),grid;
        title(['Azimuth Rate, Blind Spot Half-Width =',num2str(cone),' d
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Azimuth Rate, (deg./sec.)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Azimuth acceleration? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,azaccel*180/pi),grid;
        title(['Azimuth Accel., Blind Spot Half-Width =',num2str(cone),'
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Azimuth Acceleration, (deg./sec.^2)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
```

```
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Az. disturbance frequency? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        plot(t,fcg*azrate*N/(2*pi)),grid;
        title(['Azimuth Disturbance Frequency,',num2str(fcg),' cycles pe
        xlabel('Time Past Closest Approach to Zenith, (sec.)')
        ylabel('Frequency, (Hz.)')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end


A=input('Frequency responses? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        A=input('Open loop transfer function? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                w=logspace(-1,2);
                [mag,phase]=bode(Aznumoltf,Azdenoltf,w);
                subplot(211),loglog(w,mag),grid;
                title(['Open Loop Transfer Function, fc=',num2str(fc),'
                xlabel('Frequency, (rad./sec.)')
                ylabel('Magnitude')
                subplot(212),semilogx(w,phase-360),grid;
                xlabel('Frequency, (rad./sec.)')
                ylabel('Phase, (deg.)')
                pause
                subplot(111)
                A=input('Save the plot? Y/N [N]:','s');
                if isempty(A)
                        A='N';
                end
                if A~='N'
                        meta d:\matlab\mgn\drives\driveplot
                end
        end

        A=input('Closed loop transfer function? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
```

```
                        w=logspace(-1,2);
                        [mag,phase]=bode(Aznumcltf,Azdencltf,w);
                        subplot(211),loglog(w,mag),grid;
                        title(['Closed Loop Transfer Function, fc=',num2str(fc),
                        xlabel('Frequency, (rad./sec.)')
                        ylabel('Magnitude')
                        subplot(212),semilogx(w,phase),grid;
                        xlabel('Frequency, (rad./sec.)')
                        ylabel('Phase, (deg.)')
                        pause
                        subplot(111)
                        A=input('Save the plot? Y/N [N]:','s');
                        if isempty(A)
                                A='N';
                        end
                        if A~='N'
                                meta d:\matlab\mgn\drives\driveplot
                        end
                end

                A=input('Disturbance transmission transfer function? Y/N [N]:','
                if isempty(A)
                        A='N';
                end
                if A~='N'
                        w=logspace(-1,2);
                        [mag,phase]=bode(Aznumdj,Azdendj,w);
                        loglog(w,mag/4.85e-6),grid;
                        title(['Azimuth Disturbance Transmission, fc=',num2str(f
                        xlabel('Frequency, (rad./sec.)')
                        ylabel('Magnitude, arcsec./Nm')
                        text(.2,.9,['Transmission of drive torque disturbance to
                        text(.2,.85,['AzI=',num2str(AzI),' kg*m^2.'],'sc')
                        pause
                        A=input('Save the plot? Y/N [N]:','s');
                        if isempty(A)
                                A='N';
                        end
                        if A~='N'
                                meta d:\matlab\mgn\drives\driveplot
                        end
                end
        end
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The following section calculates the altitude response.                 %
%       The calculations are basically the same as above.                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


A=input('Work the alt. axis? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
```

```
        %%%%%%%%%%%%
Altk=(25*AltI*wc^2)/6;
Alta=5*wc;
Altb=wc/5;

Altnumoltf=[Altk 2*Altk*Altb Altk*Altb^2];
Altdenoltf=[AltI Alta*AltI 0 0 0];

Altnumdj=[1 Alta 0];
Altdendj=[AltI Alta*AltI Altk 2*Altk*Altb  Altk*Altb^2];

Altnumcltf=[Altk 2*Altk*Altb Altk*Altb^2];
Altdencltf=[AltI Alta*AltI Altk 2*Altk*Altb Altk*Altb^2];
Altta=AltI*altaccel;  %Computed motor torque applied to altitude.
Alttr=abs((pr)*Altta.*abs(sin((fr/2)*N*altpos)));  %Computed ripple torque.
Alttc=(ca)*N*(sin((fcg)*N*altpos));  %Computed cogging torque.
Alttcr=Alttc+Alttr;  %Total torque perturbations.
Altx=lsim(Altnumdj,Altdendj,Alttcr,t);

plot(t,Altx/4.85e-6),grid;
title(['Altitude Position Error, fc=',num2str(fc),' Hz.'])
xlabel('Time Past Closest Approach to Zenith, (sec.)')
ylabel('Altitude Position Error, (arcsec.)')
text(.2,.85,['Drive Ratio=',num2str(N)],'sc')
text(.2,.8,['AltI=',num2str(AltI),' kg*m^2.'],'sc')
text(.2,.75,['Cogging Amplitude per Motor=',num2str(ca),' Nm.'],'sc')
text(.2,.7,['Ripple Torque=',num2str(pr*100),'% of applied torque.'],'sc')
text(.2,.65,['Zenith Blind Spot=',num2str(cone),' deg. half-width'],'sc')
text(.2,.6,['Note: Any starting transient may be ignored.'],'sc')
text(.2,.5,['Standard Deviation=',num2str(std(Altx(500:1500)/4.85e-6)),' arcsec.
pause


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

if A~='N'
        meta d:\matlab\mgn\drives\driveplot
end


A=input('Other plots? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'

        A=input('Disturbance torques? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                plot(t,Alttr,t,Alttc),grid;
                title(['Disturbance Torque, Cogging Amp.=',num2str(ca),...
                ' Nm, Ripple=',num2str(pr*100),'%'])
                xlabel('Time Past Closest Approach to Zenith, (sec.)')
                ylabel('Disturbance Torque, (Nm)')
```

```
            pause
            A=input('Save the plot? Y/N [N]:','s');
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end

    A=input('Computed torque? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            plot(t,Altta),grid;
            title(['Computed Torque, Altitude Inertia=',num2str(AltI),' kg*m
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Computed Torque, (Nm)')
            pause
            A=input('Save the plot? Y/N [N]:','s');
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end

    A=input('Altitude position? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            plot(t,altpos*180/pi),grid;
            title(['Altitude Position, Blind Spot Half-Width =',num2str(cone
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Altitude Position, (deg.)')
            pause
            A=input('Save the plot? Y/N [N]:','s');
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end


    A=input('Altitude rate? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            plot(t,altrate*180/pi),grid;
            title(['Altitude Rate, Blind Spot Half-Width =',num2str(cone),'
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Altitude Rate, (deg./sec.)')
            pause
            A=input('Save the plot? Y/N [N]:','s');
```

```
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end

    A=input('Altitude acceleration? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            plot(t,altaccel*180/pi),grid;
            title(['Altitude Accel., Blind Spot Half-Width =',num2str(cone),
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Altitude Acceleration, (deg./sec.^2)')
            pause
            A=input('Save the plot? Y/N [N]:','s');
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end

    A=input('Alt. disturbance frequency? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            plot(t,fcg*altrate*N/(2*pi)),grid;
            title(['Altitude Disturbance Frequency,',num2str(fcg),' cycles p
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Frequency, (Hz.)')
            pause
            A=input('Save the plot? Y/N [N]:','s');
            if isempty(A)
                    A='N';
            end
            if A~='N'
                    meta d:\matlab\mgn\drives\driveplot
            end
    end

    A=input('Total sky error? Y/N [N]:','s');
    if isempty(A)
            A='N';
    end
    if A~='N'
            totptgerr=sqrt(((Altx/4.85e-6).^2+azptgerr.^2));
            plot(t,totptgerr),grid;
            title(['Sky Pointing Error, fc=',num2str(fc),' Hz.'])
            xlabel('Time Past Closest Approach to Zenith, (sec.)')
            ylabel('Pointing Error, (arcsec.)')
            text(.2,.9,['Blind Spot=',num2str(cone),' deg. half-width.'],'sc
            text(.2,.85,['Drive Ratio=',num2str(N)],'sc')
            text(.2,.8,['AzI=',num2str(AzI),' kg*m^2, AltI=',num2str(AltI),'
            text(.2,.75,['Cogging Amplitude per Motor=',num2str(ca),' Nm.'],
```

```
        text(.2,.7,['Ripple Torque=',num2str(pr*100),'% of applied torqu
        text(.2,.65,['Zenith Blind Spot=',num2str(cone),' deg. half-widt
        text(.2,.6,['Note: Any starting transient may be ignored.'],'sc'
        text(.2,.5,['Standard Deviation=',num2str(std(totptgerr(500:1500
        pause

        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end

        plot(Altx(500:2000)/4.85e-6,azptgerr(500:2000),'.');
        rms=sqrt(mean(totptgerr.^2));
        stdev=rms-sqrt(mean(Altx(500:2000)/4.85e-6).^2+mean(azptgerr(500
        title(['Sky Pointing Error, fc=',num2str(fc),' Hz.'])
        xlabel('Error due to Alt. Axis (arcsec.)')
        ylabel('Error due to Az. Azis, (arcsec.)')
        text(.2,.9,['Blind Spot=',num2str(cone),' deg. half-width.'],'sc
        text(.2,.85,['Drive Ratio=',num2str(N)],'sc')
        text(.2,.8,['AzI=',num2str(AzI),' kg*m^2, AltI=',num2str(AltI),'
        text(.2,.75,['Cogging Amplitude per Motor=',num2str(ca),' Nm.'],
        text(.2,.7,['Ripple Torque=',num2str(pr*100),'% of applied torqu
        text(.2,.65,['Zenith Blind Spot=',num2str(cone),' deg. half-widt
        text(.2,.6,['Note: Any starting transient may be ignored.'],'sc'
        text(.2,.5,['RMS Error=',num2str(rms),' arcsec.'],'sc')
        pause

        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end

end



A=input('Frequency responses? Y/N [N]:','s');
if isempty(A)
        A='N';
end

if A~='N'
        A=input('Open loop transfer function? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                w=logspace(-1,2);
                [mag,phase]=bode(Altnumoltf,Altdenoltf,w);
                subplot(211),loglog(w,mag),grid;
                title(['Open Loop Transfer Function, fc=',num2str(fc),'
                xlabel('Frequency, (rad./sec.)')
                ylabel('Magnitude')
```

```
        subplot(212),semilogx(w,phase-360),grid;
        xlabel('Frequency, (rad./sec.)')
        ylabel('Phase, (deg.)')
        pause
        subplot(111)
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Closed loop transfer function? Y/N [N]:','s');
if isempty(A)
        A='N';
end
if A~='N'
        w=logspace(-1,2);
        [mag,phase]=bode(Altnumcltf,Altdencltf,w);
        subplot(211),loglog(w,mag),grid;
        title(['Closed Loop Transfer Function, fc=',num2str(fc),
        xlabel('Frequency, (rad./sec.)')
        ylabel('Magnitude')
        subplot(212),semilogx(w,phase),grid;
        xlabel('Frequency, (rad./sec.)')
        ylabel('Phase, (deg.)')
        pause
        subplot(111)
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
end

A=input('Disturbance transmission transfer function? Y/N [N]:','
if isempty(A)
        A='N';
end
if A~='N'
        w=logspace(-1,2);
        [mag,phase]=bode(Altnumdj,Altdendj,w);
        loglog(w,mag/4.85e-6),grid;
        title(['Altitude Disturbance Transmission, fc=',num2str(
        xlabel('Frequency, (rad./sec.)')
        ylabel('Magnitude, arcsec./Nm')
        text(.2,.9,['Transmission of drive torque disturbance to
        text(.2,.85,['AltI=',num2str(AltI),' kg*m^2.'],'sc')
        pause
        A=input('Save the plot? Y/N [N]:','s');
        if isempty(A)
                A='N';
        end
        if A~='N'
                meta d:\matlab\mgn\drives\driveplot
        end
```

```
function [gamma,gammad,gammadd]=azpra(lat,cone,t)
% function [gamma,gammad,gammadd]=azpra(lat,cone,t)
% This function calcualtes the azimuth position, rate, and acceleration
% as a function of zenith angle "cone" (in degrees) as measured from
% zenith as a function of time (t is a column vector). At time=0, the
% position is at the closest approach to zenith.  This function also takes
% the local latitude "lat"  in degrees as an argument (usually 30 for CIW).
% All values are returned in radian measure.

alphad=(15/3600)*pi/180; %Earth rate in rad/sec.
alpha=alphad*t;
del=90-lat-cone; %Calculate declination.  del is the angle from the pole.
del=del*pi/180; % Convert del to radians.
lat=lat*pi/180; %Convert lat to radians.

sina=sin(alpha);
cosa=cos(alpha);
sind=sin(del);
cosd=cos(del);
cosL=cos(lat);
sinL=sin(lat);

tangamma=-(sina * sind) ./ ((sinL * cosa .* sind) - (cosL * cosd));
gamma=atan(tangamma);
cosg=cos(gamma);
sing=sin(gamma);

gammad=-(alphad .* sind) .* (sinL .* sind - cosL .* cosd) ./ ...
        ((sinL .* cosa .* sind - cosL .* cosd).^2 + (sina .* sind).^2);


gammadd=(-2 .* alphad .* (sinL .* sind - cosL .* cosd) .* sind) .* ...
        (alphad .* cosg.^2 .* sinL .* sina .* sind - gammad .* cosg .* sing .* .
        (sinL .* cosa .* sind - cosL .* cosd)) ./ ...
        ((sinL .* cosa .* sind - cosL .* cosd).^3);
```

```
function [phi,phid,phidd]=altpra(lat,cone,t)
% function [phi,phid,phidd]=altpra(lat,cone,t)
% This function calcualtes the altitude position, rate, and acceleration
% as a function of zenith closest approach angle "cone" (in degrees) as
% measured from zenith as a function of time (t is a column vector).
% At time=0, the position is at the closest approach to zenith.  This function
% takes as an argument the local latitude "lat" in degrees.  Values are
% returned in radian measure.


alphad=(15/3600)*pi/180; %Earth rate in rad/sec.
alpha=alphad*t; %Calculate hour angle.
del=90-cone-lat; %Calculate declination of target star.
lat=lat*pi/180; % Convert latitude to radians.
del=del*pi/180; % Convert declination to radians.

cosL=cos(lat);
sinL=sin(lat);
tanL=tan(lat);
cosd=cos(del);
sind=sin(del);
cosa=cos(alpha);
sina=sin(alpha);

sinphi=(cosL .* cosa .* sind) + (sinL * cosd);
phi=asin(sinphi);
cosphi=cos(phi);

phid=-(alphad .* cosL .* sina .* sind) ./ cosphi;

phidd=(alphad./(cosphi.^2)) .* ...
      (-alphad .* cosL .* cosa .* sind .* cosphi - phid .* sinphi .* cosL .* sin
```