Jason Silic

Yetneberk Worku

Final Project Report for EE 370L

**Hubble Space Telescope Compensator Analysis**

 For this final project a problem from the class’s textbook was selected for detailed analysis by the students. This report is based on problem 10.47, which examines “thermal flutter” in the Hubble Space Telescope. The telescope pointing system will have a compensator placed in cascade with the controller to damp these oscillations. We examine the frequency response of the proposed compensator and show how it will damp specific frequencies.

**Proposed compensator and MATLAB bode plot**

 The cascade compensator has a transfer function:

$$G\_{c}\left(s\right)=\frac{1.96(s^{2}+s+0.25)(s^{2}+1.26s+9.87)}{(s^{2}+0.015s+0.57)(s^{2}+0.083s+17.2)}$$

 When MATLAB is used to plot the frequency response we get the following plot, generated by the listed commands.

>> n1 = [1 1 .25]; n2 = [1 1.26 9.87]; num = 1.96\*conv(n1, n2)

>> den = conv([1 .015 .57], [1 .083 17.2]); compensator1 = tf(num, den)

compensator1 =

 1.96 s^4 + 4.43 s^3 + 22.3 s^2 + 19.96 s + 4.836

 ------------------------------------------------

 s^4 + 0.098 s^3 + 17.77 s^2 + 0.3053 s + 9.804

>> bode(compensator1)

 Note the magnitude response peaks at 0.755 and 4.15 radians/second. Also note the moderate phase response for most frequencies. We can guess that the compensator will damp the above frequencies, but there is a difficulty. The compensator is *amplifying* the two frequencies in the forward path. How can we prove this result in damping in the closed-loop transfer function?

**Derivation of complete transfer function and MATLAB simulation**

 The assumed block diagram for the telescope pointing control is shown below. The primary controller transfer function is G(s), the compensator is GC(s), and the plant is GP(s). Unity feedback is assumed. The thermal disturbance is added to the control loop as D(s).

**Block diagram of pointing control system**



 We can define an error signal as E as the input to the compensator:

E = R – Y Therefore: Y = R – E

Y = DGp + EGcGGp Substituting into the first equation: E = R - DGp - EGcGGp

Combining factors of E: E(1+GcGGp) = R – DGp 🡪 $E=R\frac{1}{1+G\_{c}GG\_{p}}-D\frac{G\_{p}}{1+G\_{c}GG\_{p}}$

Substituting into Y = R – E:

$Y= R-R\frac{1}{1+ G\_{c}GG\_{p}}+D\frac{G\_{p}}{1+ G\_{c}GG\_{p}}$

Multiply by R by ($1+G\_{c}GG\_{p}$)/ ($1+G\_{c}GG\_{p}$), combine terms, and simplify:

$$Y= R\frac{G\_{c}GG\_{p}}{1+ G\_{c}GG\_{p}}+D\frac{G\_{p}}{1+ G\_{c}GG\_{p}}$$

 We therefore see that the transfer function from the disturbance to the output is constructed with the compensator transfer function in the denominator. This means that large gain at the critical frequencies of .755 and 4.15 radians/second will significantly damp those frequencies.

MATLAB can be used to substantiate these claims. We assume a unity transfer function for the

plant and control system for testing purposes and find the frequency response of the function $\frac{1}{1+G\_{c}}$. The results with other transfer function are similar, as will be shown at the end of the report. The MATLAB code is listed, followed by the bode plot. Note that we have multiplied to make it easy for MATLAB:

$$\frac{1}{1+\frac{num}{den}}=\frac{den}{den+num}$$

>> tr\_feedback = tf(den, (num + den))

tr\_feedback =

 s^4 + 0.098 s^3 + 17.77 s^2 + 0.3053 s + 9.804

 --------------------------------------------------

 2.96 s^4 + 4.528 s^3 + 40.08 s^2 + 20.27 s + 14.64

>> bode(tr\_feedback)



 We now have attenuation of the listed frequencies. Gain values of -38.1dB and -34.7dB provide significant damping at the frequencies of the “thermal flutter.” This is confirmed when we examine the original paper referenced in the textbook. The thermally induced oscillations have frequencies of 0.12Hz and 0.66Hz. This corresponds to 0.7540 and 4.147 radian/second, respectively. (the conversion factor is 2π; there are 2π radians in a revolution) These frequencies are very close (essentially identical if you take rounding into account) to the frequencies that MATLAB reports for the peak damping. We therefore have experimental confirmation of the damping of these oscillations by the proposed compensator.

**Simulation with non-unity controller and plant**

 We would like to confirm our observations if we use possible real transfer functions for the

pointing controller and plant. We choose the controller G(s) and the plant Gp(s) to have the following

transfer functions:

$G\left(s\right)=\frac{8}{s+8}$ $G\_{p}\left(s\right)=\frac{125}{s^{2}+10s+125}$

 The controller has a pole at s = -8, and the plant has poles at s = -5 ±j10. The MATLAB code again uses an equivalent formula for easy computation. We find the transfer function for the disturbance:

T(s) = Y(s)/D(s) = $\frac{G\_{p}}{1+G\_{c}GG\_{p}}$

>> nc = 1.96\*conv([1 1 .25], [1 1.26 9.87]);

>> dc = conv([1 .015 .57], [1 .083 17.2]);

>> n = [8]; d = [1 8]; np = [125]; dp = [1 10 125];

>> num = conv(conv(np, dc), d); den1 = conv(nc, conv(n, np));

>> den2 = conv(dc, conv(d, dp));

>> size(den2) ans = 1 8

>> size(den1) ans = 1 5

>> den1 = [0 0 0 den1]

den1 = 1.0e+04 \*

 0 0 0 0.1960 0.4430 2.2305 1.9963 0.4836

>> den = den1+den2

den = 1.0e+04 \*

 0.0001 0.0018 0.0225 0.3300 0.8186 4.0315 2.2278 1.4640

>> disturbance\_tr = tf(num, den)

disturbance\_tr =

 125 s^5 + 1012 s^4 + 2319 s^3 + 1.781e04 s^2 + 1531 s + 9804

 -------------------------------------------------------------------------------

 s^7 + 18.1 s^6 + 224.5 s^5 + 3300 s^4 + 8186 s^3 + 4.032e04 s^2 + 2.228e04 s

 + 1.464e04

>> bode(disturbance\_tr)

 Because the compensator is added as a multiplier in the denominator to the transfer function, it will damp the frequencies of .755 and 4.15 rad/s regardless of what the actual plant and controller transfer functions are. As shown by the preceding plot, this is confirmed by MATLAB simulation results.

 The Nyquist plot of the compensator transfer function is very interesting. The four poles of the function are all very close to the imaginary axis, possibly leading to the loops in the Nyquist diagram. All the zero of the transfer function are also in the left-half plane, however, so the transfer function never becomes unstable at any value of gain (except negative, of course).

[Code edited to save space.]

>> tf2 = tf(nc, dc)

>> pole(tf2)

ans = -0.0415 + 4.1471i -0.0415 - 4.1471i

 -0.0075 + 0.7549i -0.0075 - 0.7549i

>> nyquist(tf2)

 When plotting the Nyquist diagram, s = jω for part of the plot. Note that s will approach very close to the poles in the compensator transfer function that are barely in the left-half plane, as can be seen in the listing above. The magnitude of the mapped point will therefore increase at frequencies near these poles. This is confirmed when the MATLAB diagram is examined and will be discussed during the presentation.

**Conclusion**

 This exercise has provided the student with an excellent opportunity to examine a real problem and apply theoretical analysis. MATLAB proficiency has also been improved, and a greater appreciation for the power of that software tool is inevitable.