

Attitude Tracking Control Simulating the Hubble Space Telescope

Brett Streetman

December 12, 2003

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction and Background | 1 |
| 1.1 | Hubble Space Telescope | 1 |
| 1.1.1 | Pointing Control System Operations | 2 |
| 1.2 | Attitude Tracking Control | 3 |
| 2 | Problem Statement and Equations of Motion | 4 |
| 3 | Analysis | 6 |
| 3.1 | Reference Motion | 6 |
| 3.2 | PID Controller | 7 |
| 3.2.1 | Initial Gain Selection | 8 |
| 4 | Results | 10 |
| 4.1 | Linear and Nonlinear Equations | 10 |
| 4.2 | Static Gains | 11 |
| 4.3 | Gain Scheduling | 12 |
| 4.4 | Continuous State-Dependent Gains | 14 |
| 4.5 | Conclusion | 16 |

Chapter 1

Introduction and Background

The Hubble Space Telescope (HST) was launched April 24, 1990 in the shuttle Discovery. The HST was named after American astronomer Edwin Hubble, made famous for his discovery of the constant expansion of the universe. The telescope bearing his name has made some of the most important astronomical observations of the modern era. The Hubble was design for versatility and longevity. It has perform many diverse missions, including documenting the structure of the Solar System, searching for planets in other solar systems, and attempting to discover the size and nature of the entire Universe.

To take scientifically useful photos, the attitude control of the Hubble must be stringent. The design criteria are for it to be able to track an inertially fixed target with error of no more than 0.007 arcseconds for anywhere from 10 minutes to 24 hours[1]. The accuracy involved is equivalent to hitting a dime with laser from 322 kilometers away and never straying past the edge of the coin.

1.1 Hubble Space Telescope

Plans for the eventual space telescope began at NASA in the early 1970's. Originally, the telescope was supposed to have a 3 meter aperture, but that was later relaxed to 2.4 meters[2]. The size reduction went hand in hand with other specification relaxations to reduce the cost of the project in a 1974 redesign. In 1976, Glaese, et al.[2] presented a Pointing Control System (PCS) that much resembles the system used on the Hubble. Dougherty, et al.[3] produced a more detailed document on the PCS hardware in 1982. The proposed testing procedure for the PCS was outlined in this paper, and called for the construction of a spherical air bearing tabletop simulator to test the fine pointing scheme.

After the Hubble was launched, vibrations in the solar panels caused the Pointing Control System (PCS) to not perform up to specifications. In 1995, NASA scientists Foster, et al.[4] described the nature of these vibrations. The excitations were ascribed to thermal deformations of the solar panels. The vibrations were more intense than

predicted by the attitude system designers and the PCS could not overcome them. The initial controller was redesigned and uploaded to the Hubble, but NASA took advantage of the large amount of data collected to solve the problem and commissioned five different organizations to develop their own controller for HST. Addington and Johnson[5] at the University of Alabama developed a dual-mode disturbance-accommodating controller that both provides torques to counteract disturbances and torques to augment the natural damping of the solar arrays. Collins and Richter[6] at Florida A&M University created a linear quadratic Gaussian design for the PCS. An H_∞ controller was both analytically and numerically derived by Irwin, et al.[7] at Ohio University. Zhu, et al.[8] developed a controller for the Hubble based on covariance control techniques. Nurre, et al.[9] from NASA's Marshall Spaceflight Center described the evolution of the SAGA-II controller. The SAGA-II was an extension of the original controller on the Hubble, designed to account for solar array vibrations. On April 16, 1992, the SAGA-II controller became the normal operating mode for the Hubble, and succeeded in reducing the effect of the vibrations to an acceptable level.

In addition to its initial attitude problems, Hubble had an aberration on one of its mirrors that caused low picture quality. A corrective optics unit was installed on the telescope during its first servicing mission. The Hubble could begin its science work in earnest now.

With its initial problems fixed, the Hubble began to take photographs that met and exceeded original expectations. In 1996, Fischer and Duerbeck[10] published a book displaying a large number of these photos. Along with the large number of photographs, the book contained a full description of Hubble's systems and an overview of the science missions being pursued by the HST.

The official NASA HST website[1] has the most current information on the Hubble. This includes specifications, subsystem descriptions, service timelines, and mission objectives. Much information, current or historical, can be found on this website. A second website with more technical information is located at Ref. [11].

1.1.1 Pointing Control System Operations

Pointing Control System operation begins with coarse attitude determination. The HST contains a sun sensor and magnetometer to get a first approximation of its orientation. The Fixed Head Star Trackers then search the starfield for recognized patterns to gain more exact attitude knowledge. The Hubble is reoriented by the use of four reaction wheels. The HST contains six rate gyros. Three of these are operating at any given time to determine the angular velocities of the telescope. When the Hubble has acquired its target using the "coarse" guidance of the Fixed Head Star Trackers, the fine guidance system takes control. The Fine Guidance Sensors (FGS) are used to gain a near exact lock on a target. There are three FGSs on the Hubble. Two lock onto individual stars around the target, while the third focuses itself and

the optics onto the target itself. The two FGSs not on the target keep their individual star inside of a fixed area in their field of view, using the information from the rate gyros and controlling with the reaction wheels.

1.2 Attitude Tracking Control

Pointing HST at an astronomical target involves the Hubble tracking that object for a certain period of time. Most of Hubble's target are many light years away, so they can be considered fixed in the inertial reference frame. Hablani[12] has performed a large amount of work in this area, extending many principles to the tracking of non-inertially fixed targets. Hall[13] has also worked with attitude tracking and has developed a tracking controller using momentum wheels and thrusters.

Chapter 2

Problem Statement and Equations of Motion

The attitude motion required by the Hubble Space Telescope will be examined. The HST must track an inertially fixed object as it propagates through its orbit. With the target fixed in space, the body axes of the Hubble must remain at a constant orientation with respect to the inertial frame. To simplify the analysis of this motion, several assumptions will be made. The motion will be simulated for one half of an orbital period. The body frame and the orbital frame will be initially aligned. The target will lie at an infinite distance away in the plane of the orbit. These assumptions are displayed in Figure 2.1. The target lying in the orbital plane creates a situation in which only pitch motion is significant. If the target were not in plane, there would be roll and yaw motion, but they would be very similar to the pitch motion. Removing roll and yaw motion from the problem does not remove any interesting dynamics, just another gravity gradient torque about a different axis.

Pitch-only motion allows for the use of the linear and nonlinear forms of the decoupled pitch equation. The pitch equations are derived using a 1-2-3 Euler angle sequence with θ_2 as the pitch angle and include the effects of gravity gradient torque. These equations are taken from Ref. [14]. The nonlinear pitch equation is given by:

$$I_2\ddot{\theta}_2 + 3\omega_c^2(I_1 - I_3)\sin\theta_2\cos\theta_2 = u \quad (2.1)$$

where I_n is the principle moment of inertia about the n th body axis and u is the applied control torque. The pitch equation can be linearized to yield the simpler

$$I_2\ddot{\theta}_2 + 3\omega_c^2(I_1 - I_3)\theta_2 = u \quad (2.2)$$

These two equations will represent the dynamics of the system.

A proportional, integral, derivative (PID) controller will be used to force the Hubble to track the fixed target. A PID controller is a linear controller so it will be derived based on the linear pitch equation, Eq. 2.2. The controller will produce three gains; a proportional gain, k_p , an integral gain, k_i , and a derivative gain, k_d .

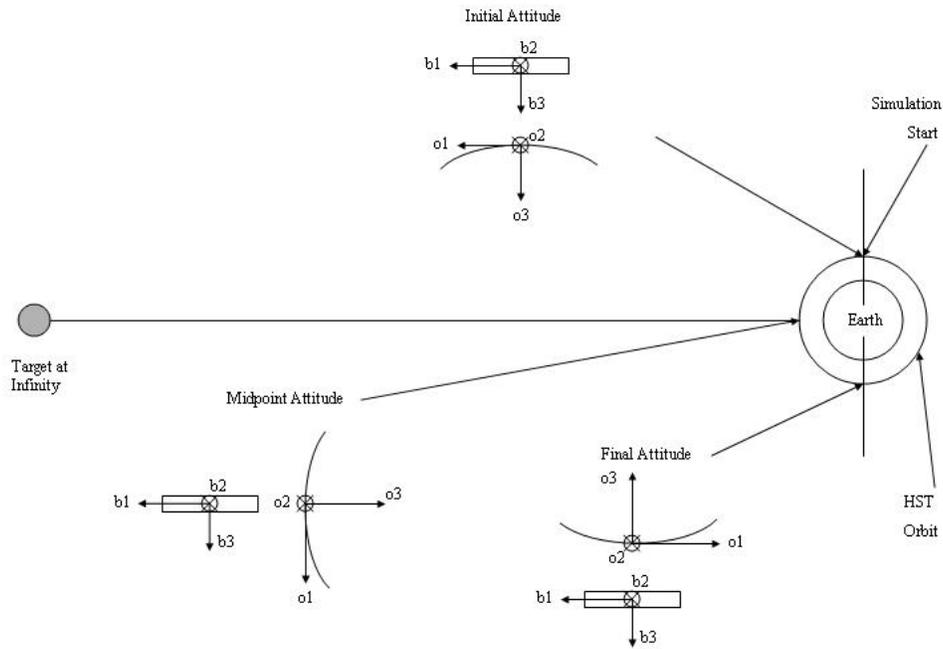


Figure 2.1: Attitude motion to be simulated

These three gains will be tuned to work with the actual reaction wheel hardware on HST. The gains will be tuned in three different ways: a static gain case, a scheduled gain case, and a continuous state-dependent gain case.

Chapter 3

Analysis

3.1 Reference Motion

The desired motion of the Hubble can be described by a virtual spacecraft that perfectly tracks the target. The controller will then try to match this motion through time. The desired motion will be defined by a desired pitch angle, θ_d . The controller operates by minimizing the error between θ_d and θ_2 .

The first step in determining the reference motion is to determine Hubble's orbit. A recent two line element set (TLE) for HST was obtained from Ref. [15]. The TLE was analyzed to calculate the orbital elements of the orbit. Using the orbital elements, the rotation matrix between the orbital frame and the inertial frame, R^{oi} , can be calculated by an equation from Ref. [14]:

$$R^{oi} = \begin{bmatrix} -su \, c\Omega - cu \, ci \, s\Omega & -su \, s\Omega + cu \, ci \, c\Omega & cu \, si \\ -si \, s\Omega & si \, c\Omega & -ci \\ -cu \, c\Omega + su \, ci \, s\Omega & -cu \, s\Omega - su \, ci \, c\Omega & -su \, si \end{bmatrix} \quad (3.1)$$

where s or c denotes sin or cos of an angle, and i is the inclination, Ω is the right ascension of the ascending node, and u is the argument of latitude. Based on the assumptions made for this problem, the orbital frame and the body frame are initially aligned, so

$$R^{bi} = R^{oi}|_{t=0} \quad (3.2)$$

The body frame is constant with respect with to the inertial frame so this R^{bi} is valid for all time.

A forward propagation of the orbital elements can then be performed. The propagation was performed using a Matlab code obtained from Ref. [14]. The time step chosen for this propagation was 0.5 seconds. After each step forward, a new set of orbital elements was plugged into Eq. 3.1. The new R^{oi} was used with R^{bi} to calculate the rotation between the body frame and the orbital, R^{bo} , by:

$$R^{bo} = R^{bi}(R^{oi})^T \quad (3.3)$$

The orbital to body rotation matrix for a 1-2-3 Euler angle sequence takes the following form[14]:

$$R^{bo} = \begin{bmatrix} c_2c_3 & s_1s_2c_3 + c_1s_3 & s_1s_3 - c_1s_2c_3 \\ -s_2c_3 & c_1c_3 - s_1s_2s_3 & s_1c_3 + c_1s_2s_3 \\ s_2 & -s_1c_2 & c_1c_2 \end{bmatrix} \quad (3.4)$$

where s_n and c_n represents the sin or cos of θ_n . The (3,1) element of the R^{bo} matrix depends only on the pitch angle. The desired pitch angle, θ_d can be easily solved for by:

$$\theta_d = \sin^{-1}(R^{bo}(3, 1)) \quad (3.5)$$

The correct quadrant of θ_d can be found by combining other terms of R^{bo} to get another expression for the pitch angle. The resulting virtual satellite reference motion is presented in Fig. 3.1. As can be easily seen in the figure, θ_d increases linearly with time. Fitting a line to this data yields the result of

$$\theta_d = \frac{2\pi}{T_p}t \quad (3.6)$$

where T_p is the period of Hubble's orbit. The linear nature of this result removes several difficulties in control that arise in a reference motion that is nonlinear and not known as a discrete function of time.

3.2 PID Controller

A proportional, integral, derivative (PID) controller is a simple type of linear controller. A PID controller is not complicated to derive and not difficult to implement. The interesting part of PID control comes in the selection of its three gains: the proportional gain, k_p , the integral gain, k_i , and the derivative gain k_d .

A PID controller assumes a control of the form

$$u = -k_p x - k_i \int_0^t x(\tau) d\tau - k_d \dot{x} \quad (3.7)$$

where x is the state that is to be controlled. In this problem, the pitch angle of the actual spacecraft is supposed to be tracking the desired pitch angle of the virtual spacecraft. The state to be controlled then is defined as $\delta\theta$, which is given by

$$\delta\theta = \theta_d - \theta_2 \quad (3.8)$$

where θ_2 is the pitch angle of the actual spacecraft. The PID control term can now be written as

$$u = -k_p \delta\theta - k_i \int_0^t \delta\theta d\tau - k_d \dot{\delta\theta} \quad (3.9)$$

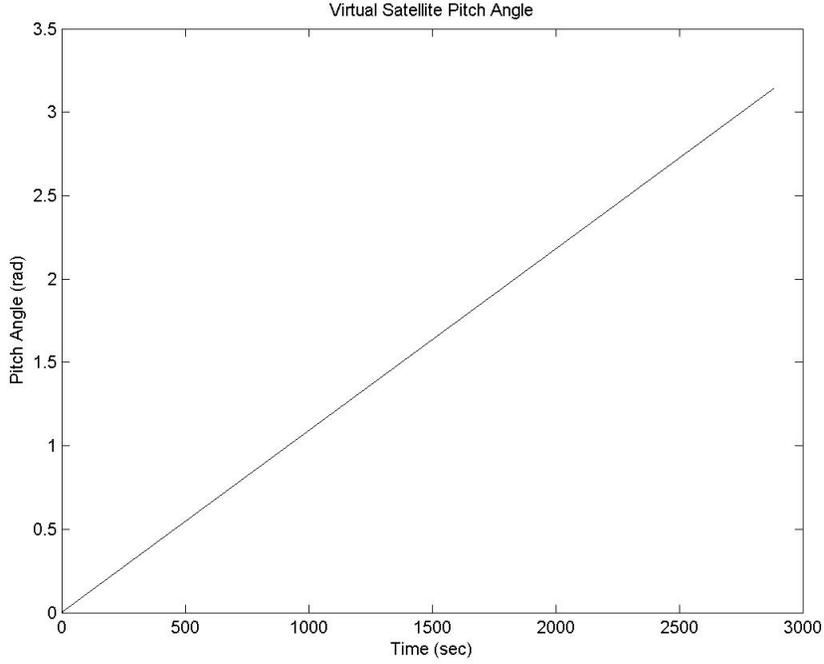


Figure 3.1: Reference motion pitch angle

The integral and derivative of θ_d can easily be calculated from Eq. 3.6 and the values for θ_2 are part of the states and must be integrated with the equations of motion. The linear equation of motion now becomes

$$I_2\ddot{\theta}_2 + 3\omega_c^2(I_1 - I_3)\theta_2 + k_p\delta\theta + k_i\int_0^t \delta\theta d\tau + k_d\dot{\delta}\theta = 0 \quad (3.10)$$

A PID controller can easily be shown to be a stable controller by forming the Laplace transform characteristic equation and applying the Final Value Theorem to it.

The PID controller is applied to the nonlinear equation of motion by simply inserting the control torque expression of Eq. 3.9 into the nonlinear pitch equation.

3.2.1 Initial Gain Selection

An initial guess at the proper gains can be found by factoring the characteristic equation of the system in terms of three new variables: the natural frequency, ω_n , the damping ratio, ζ , and the integral time constant, T . The gains are classically defined by these terms as[14]:

$$k_{p0} = \omega_n^2 + \frac{2\zeta\omega_n}{T} - k_o \quad (3.11)$$

$$k_{i_0} = \frac{\omega_n^2}{T} \quad (3.12)$$

$$k_{d_0} = 2\zeta\omega_n + \frac{1}{T} \quad (3.13)$$

where the zero subscript on the gains is the notation used to reference these base guesses. In this problem, the three independent variable were defined as follows

$$\omega_n = \left(3\omega_c^2 \frac{I2 - I1}{I3}\right)^{\frac{1}{2}} \quad (3.14)$$

$$\zeta = 0.7 \quad (3.15)$$

$$T = \frac{10}{\zeta\omega_n} \quad (3.16)$$

where ω_c is the mean motion of Hubble's orbit. These values are completely arbitrary and are only used as a starting point for tuning the gains. However, starting with reasonable values in the above equations can usually give gains within one or two orders of magnitude of the proper ones.

Chapter 4

Results

The PID controller presented in the last chapter was applied to a variety of cases with both the linear and nonlinear equations of motions in a Matlab simulation. The gains were also tuned in a variety of ways. First, a single set of gains was used for the entire length of the simulation. Next, the method of gain scheduling was applied. Gain scheduling involves the use of two or more discrete sets of gains, depending on whether the error is large or small. The third method applied is referred to as continuous state-dependent gains (CSDG). Implementing a CSDG controller involves choosing the gains as a continuous function of the states.

In an effort to make the simulations more realistic a maximum control torque value was implemented. The maximum torque HST could apply to any control axis was estimated at 1.6 N-m. Although this value is much larger than most satellites would have, the 2-axis moment of inertia for the Hubble is 77,217 kg-m²[9], requiring quite a large torque to get it moving. Any time a controller asked for a torque that was too large it was only allowed to apply a torque 80% of the maximum value. The maximum angular velocity of the simulated HST was also limited. The real Hubble has a maximum slew rate of 0.00175 rad/s, or about the rate of the minute hand on a clock. In simulation, this maximum angular velocity was assumed to be when the reaction wheels had reached their maximum spin rate. If the wheel is already at its maximum speed, no further control torque can be applied. This situation was modeled by having the wheels apply a small torque in the opposite direction if the maximum angular velocity was reached.

4.1 Linear and Nonlinear Equations

A Matlab simulation was developed for both the linear and nonlinear equations of motion. These simulations were run for a variety of initial conditions and gains. All of the trials show one result: the linear and nonlinear equations produce extremely similar results. This results holds even for clearly non-small angle initial conditions. Figure 4.1 shows the results of the linear and nonlinear simulations for an

initial pitch angle of 90° and an initial angular velocity of -0.0005 rad/s. The reference motion starting conditions are 0° pitch angle and 0.0011 rad/s angular velocity. The two simulations produce results that are virtually indistinguishable. The two different equations agreed similarly in every trial run. For this reason, all subsequent work was performed only on the linear equation of motion.

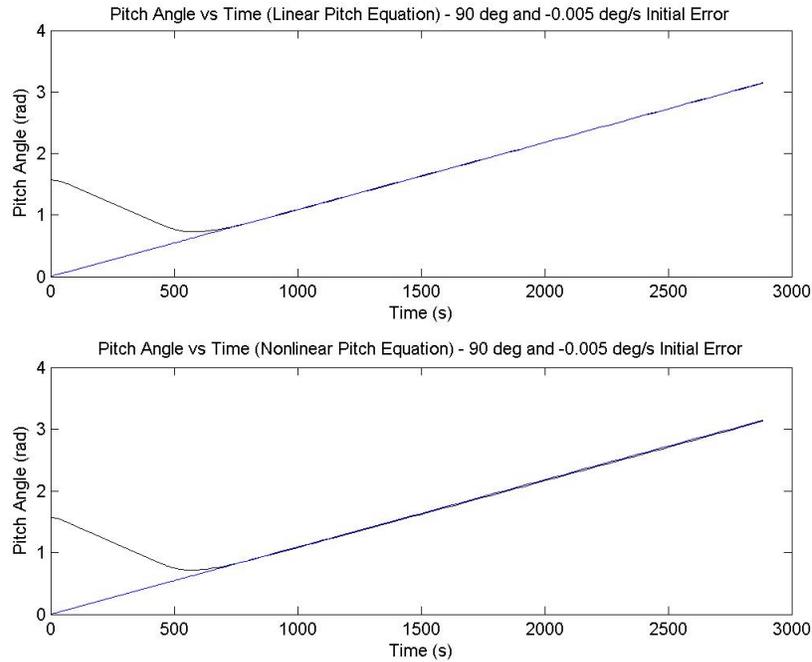


Figure 4.1: Comparison of linear and nonlinear results

4.2 Static Gains

The use of static gains is the simplest form of PID control. One set of gains is chosen at the beginning of the simulation, and that set of gains is used no matter what the states are. The gains can then be tuned in order to find the best control. In this study, the best control was defined by the fastest acquisition of the reference motion with a little overshoot as possible and the reaction wheels not reaching their maximum speed.

The first gains to try have been previously defined as k_{p0} , k_{i0} , and k_{d0} . When these gains were applied, the system was slow to respond and sometimes unable to overcome the initial error before the end of the simulation. An increase in the gains was in order. First, the proportional gain was increased by adding a multiplier to k_{p0}

until the system easily acquired the desired state. Then the integral gain was increased to facilitate a faster reduction of error. The derivative gain can then be increased to eliminate or reduce any overshoot and overly large controls introduced by the previous gain increases. This process, of course, must be iterated many times, as changing one gain changes the relative behavior of the others. The eventual optimal gain set will have the proportional and integral gains as large as possible and the derivative gain as small as possible, while still staying within the system constraints.

The constraints for this system were the maximum torque able to be applied and the maximum wheel speed, which is defined by maximum angular velocity. The torque constraint was internally applied. If a controller asked for more than the maximum torque, only the maximum value was given. If the maximum angular velocity was reached a counter-torque was applied. However, the the tuning was performed so that this never occurred. The gains for all three selection techniques were tuned for a reference set of initial conditions, namely a 30° pitch angle error and an initial angular velocity of -0.0005 rad/s. The gains were defined in terms of a multiplier on the base gains. For the static gain case best set of gains found was defined by:

$$k_p = 75k_{p0} \quad (4.1)$$

$$k_i = 50k_{i0} \quad (4.2)$$

$$k_d = 20k_{d0} \quad (4.3)$$

The result of applying these gains is presented in Fig. 4.2. The top plot shows the pitch angle of the actual spacecraft approaching the reference motion. The middle plot is of the pitch rate of the the actual craft. This quantity is the same as the angular velocity of the craft for the planar motion in the simulation. The bottom plot shows the control torque output by the controller as a function of time. The control for this case spends a short time at the minimum value before increasing rapidly, then beginning a steady decent to produce the steady reference motion. The angular velocity decreases until it just flirts with the maximum value, and then it quickly approaches the reference value. The pitch angle error rapidly decreases until reaching its desired valued with little to no overshoot.

4.3 Gain Scheduling

Gain scheduling (GS) is the next step up from static gain selection. To implement GS, multiple sets of static gains are chosen and implemented in different situations. At a minimum two set of gains are chosen, one for large errors and one for small errors. This simple implementation is what is pursued here. One set of gains was used for errors larger than half of the initial error and one set was used for errors less than half of the initial error.

Gain scheduling is an improvement over static gains because the gains can be better tailored to individual situations. When the error is large, the integral gain should

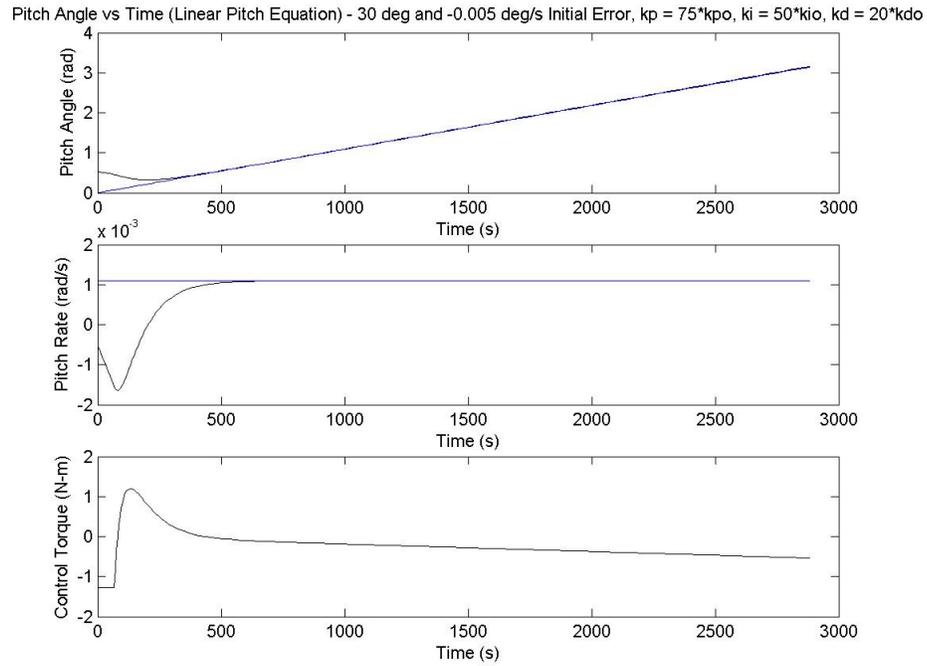


Figure 4.2: Best gain behavior for a set of static gains

be larger and the derivative gain smaller to quickly make up ground. When the error is small, the integral gain should be smaller and the derivative gain larger to reduce overshoot. In the equation of motion, the control output by the proportional control is already changing with the state, so it can usually remain constant throughout.

The two sets of gains chosen in this study were tuned to cover the error more quickly than the static gain case and still not overshoot. The best set of gains found is presented in Table 4.1. The results produced by these sets of gains are shown

Table 4.1: Best gains found by the gain scheduling method

| | Large $\delta\theta$ | Small $\delta\theta$ |
|-------|----------------------|----------------------|
| k_p | $75k_{p0}$ | $75k_{p0}$ |
| k_i | $54k_{i0}$ | $44k_{i0}$ |
| k_d | $18k_{d0}$ | $20k_{d0}$ |

in Fig. 4.3. The pitch angle data once again shows a fast, well-behaved controller. However, the GS controller reaches the reference motion 20 to 30 seconds earlier than the static controller. This improvement in speed comes at a price though. The control torque switches quickly from its minimum to its maximum value. This change is more

abrupt than seen in the static gain case and may be detrimental to a real motor. The angular velocity of the spacecraft reaches its proper value slightly faster in the GS case, but also spends more time near its maximum value. Overall, the gain scheduling controller works faster but at a higher cost.

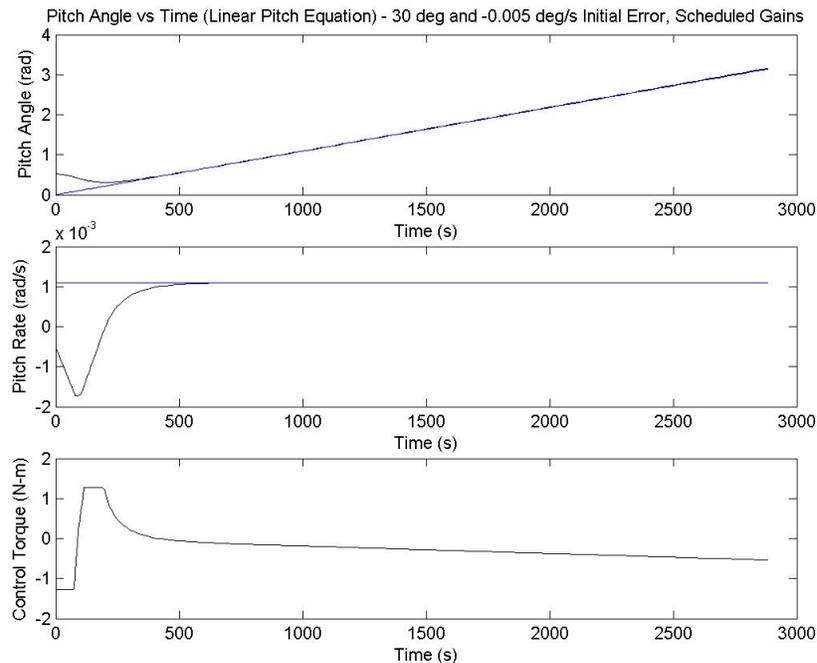


Figure 4.3: Best gain behavior for a set of scheduled gains

4.4 Continuous State-Dependent Gains

The use of continuous state-dependent gains (CSDG) takes gain scheduling to the next level. The gains are defined by a continuous function of the state, amounting to an infinite number of GS sets. The gains can then be completely and easily tailored to any situation that the dynamics dictate. However, making the gains a function of the states makes the PID controller completely nonlinear. The guarantee of PID stability no longer applies, and no proof is presented here claiming stability. Through the experiences of this study, though, it seems properly chosen gain functions not only increase the performance of the controller, but also its stability.

The gain functions are completely open and arbitrary. A few guidelines do exist. A gain function should always depend on the magnitude of the state, as negative gains are a fast track to instability. The proportional gain can once again remain constant. Its control output is already a function of the state.

The first thought for gain functions would be linearly increasing with error integral gain and linearly decreasing with error derivative gain. The gains would then be of the form:

$$k_p = 75k_{p0} \quad (4.4)$$

$$k_i = \left((k_{i_{max}} - k_{i_{min}}) \left| \frac{\delta\theta}{\delta\theta_0} \right| + k_{i_{min}} \right) k_{i0} \quad (4.5)$$

$$k_d = \left(k_{d_{max}} - (k_{d_{max}} - k_{d_{min}}) \left| \frac{\delta\theta}{\delta\theta_0} \right| \right) k_{d0} \quad (4.6)$$

where $\delta\theta_0$ is the initial error. This term is included so the state variable in the gain function usually only varies between 0 and 1. Using gain functions of this form had poor results. The controller more often than not led to instabilities in the results. Obviously new forms of the gain functions were required.

By observing the dynamics of the static and gain scheduling cases, the period of largest control changes was found to be when the error was about halfway between the initial value and zero. The derivative gain should largest at this most volatile point to rein in the integral control. The derivative gain function should therefore be maximum for a scaled state of 0.5 and smaller at the edges. A half sine wave exhibits these properties and was chosen to fulfill this role. The derivative gain function then takes the following form:

$$k_d = \left[(k_{d_{max}} - k_{d_{min}}) \sin \left(\pi \left| \frac{\delta\theta}{\delta\theta_0} \right| \right) + k_{d_{min}} \right] k_{d0} \quad (4.7)$$

A derivative gain of this form was used with a constant proportional gain and an integral gain of the form of Eq. 4.5. The results of this controller were excellent in all respects. In fact, it was thoroughly difficult to push this controller near the performance limits. The final tuning results are expressed as the following gain functions:

$$k_p = 75k_{p0} \quad (4.8)$$

$$k_i = \left(65 \left| \frac{\delta\theta}{\delta\theta_0} \right| + 30 \right) k_{i0} \quad (4.9)$$

$$k_d = \left[6.8 \sin \left(\pi \left| \frac{\delta\theta}{\delta\theta_0} \right| \right) + 12 \right] k_{d0} \quad (4.10)$$

The integral gain function shows an astounding maximum of $95k_{i0}$ compared with $54k_{i0}$ for the GS case. The derivative gain is also much smaller than either the static or GS cases. The controller, however, outperforms both other cases. The results of its control are shown in Fig. 4.4. The controller reduces the error to zero 20 to 30 seconds faster than the GS case with only a slight angular velocity overshoot. The angular velocity only briefly closes in on its maximum value. The control torque begins at its minimum value, but never reaches its maximum. After the local maximum, there is

another slight maximum that drives the error quickly to zero. This feature does not appear in the linear controller cases. The continuous state-dependent gain controller outperforms both the static gain and gain scheduling controllers, but has the danger of becoming unstable if gain functions are not properly chosen.

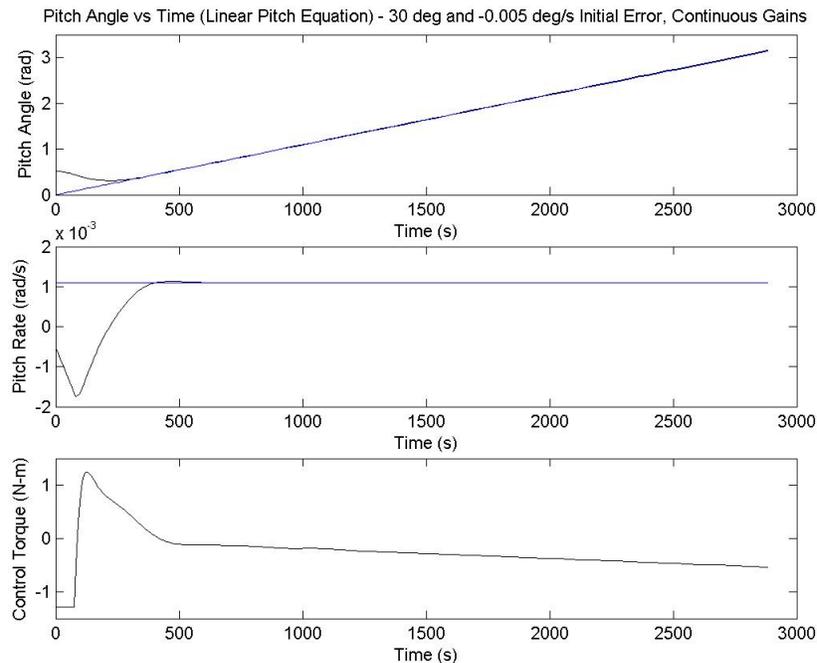


Figure 4.4: Best gain behavior for continuous state-dependent gains

4.5 Conclusion

Three types of PID control were implemented in an attitude problem inspired by the Hubble Space Telescope. A satellite tracked an inertially fixed object through the period of half an orbit. A static gain PID controller was tuned to perform this maneuver and performed well in controlling the dynamics. The static gain controller was then extended to a PID controller with gain scheduling. The gain scheduled controller performed better than the static controller, but with higher cost and more hardware wear. The gain scheduling controller was extended to a continuous state-dependent gain controller (CSDG). A continuous state-dependent gain controller has gains that are a function of the error, and thus becomes a fully nonlinear controller. The CSDG controller outperforms both the static and gain scheduled ones. However, if the gain functions are improperly selected, the CSDG controller can quickly become unstable.

Several important matters were not covered in this project. Real motors have limits of how fast they can change torque values. The controller should limit step-to-step torque changes to some reasonable value. The controller were all tuned at the same initial conditions, but this set of conditions was not the worst. The gain values presented most likely will fail at the worst case initial conditions. Finally, a stability analysis of CSDG controller should be performed. Their control is either excellent or awful and a study of when each case occurs should be performed.

Bibliography

- [1] NASA. The Hubble Space Telescope Project. Internet document, 2003. <http://hubble.nasa.gov>.
- [2] J.R. Glaese, H.F. Kennel, G.S. Nurre, S.M. Seltzer, and H.L. Shelton. Low-Cost Space Telescope Pointing Control System. *Journal of Spacecraft and Rockets*, 13(7), 1976.
- [3] H. Dougherty, K. Tompetrini, J. Levinthal, and G. Nurre. Space Telescope Pointing Control System. *Journal of Guidance, Control, and Dynamics*, 5(4), 1982.
- [4] C.L. Foster, M.L. Tinker, G.S. Nurre, and W.A. Till. Solar-Array-Induced Disturbance of the Hubble Space Telescope Pointing System. *Journal of Spacecraft and Rockets*, 32(4):634–644, 1995.
- [5] S.I. Addington and C.D. Johnson. Dual-Mode Disturbance Accommodating Pointing Controller for Hubble Space Telescope. *Journal of Guidance, Control, and Dynamics*, 18(2), 1995.
- [6] E.G. Collins and S. Richter. Linear-Quadratic-Gaussian-Based Controller Design for Hubble Space Telescope. *Journal of Guidance, Control, and Dynamics*, 18(2), 1995.
- [7] R.D. Irwin, R.D. Glenn, W.G. Frazier, D.A. Lawrence, and R.F. Follett. Analytically and Numerically Derived H_∞ Controller Designs for Hubble Space Telescope. *Journal of Guidance, Control, and Dynamics*, 18(2), 1995.
- [8] G. Zhu, K.M. Grigoriadis, and R.E. Skelton. Covariance Control Design for Hubble Space Telescope. *Journal of Guidance, Control, and Dynamics*, 18(2), 1995.
- [9] G.S. Nurre, J.P. Sharkey, J.D. Nelson, and A.J. Bradley. Preservicing Mission, On-Orbit Modification to Hubble Space Telescope Pointing Control system. *Journal of Guidance, Control, and Dynamics*, 18(2), 1995.

- [10] Daniel Fischer and Hilmar Duerbeck. *Hubble: A New Window to the Universe*. Copernicus, New York, 1996.
- [11] NASA Goddard. Hubble Space Telescope Systems. Internet document, 2003. http://www.gsfc.nasa.gov/gsfcservice/gallery/fact_sheets/spacesci/hst3-01/hubble_space_telescope_systems.htm.
- [12] H.B. Hablani. Design of a Payload Pointing Control System for Tracking Moving Objects. *Journal of Guidance, Control, and Dynamics*, 12(2), 1989.
- [13] C.D. Hall, P. Tsiotras, and H. Shen. Tracking Rigid Body Motion Using Thrusters and Momentum Wheels. *Journal of the Astronautical Sciences*, 50(3), 2002.
- [14] C.D. Hall. *Spacecraft Attitude Dynamics and Control*. Class Notes - AOE 5984, 2003.
- [15] T.S.Kelso. Celestrak. Internet document, 2003. <http://celestrak.com/>.