State Space Model of a Light Weight Flexible Telescope

A Thesis
Presented to the Faculty of
California Polytechnic State University, San Luis Obispo

In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In Mechanical Engineering

By
Chi-Yeh Hsu
August 2010
Committee Membership

Title: State Space Model of Light Weight Flexible Telescope

Author: Chi-Yeh Hsu

Date Submitted: August, 2010

Committee Chair: Dr. John Ridgely
Assistant Professor
Cal Poly Mechanical Engineering

Committee Member: Dr. Charles Birdsong
Associate Professor
Cal Poly Mechanical Engineering

Committee Member: Dr. Xi Wu
Assistant Professor
Cal Poly Mechanical Engineering
Abstract

State Space Model of a Light Weight Flexible Telescope

Chi-Yeh Hsu

The accuracy of optical telescope in the past was dominated by the construction of the telescope body and the precision of optical surface. This thesis fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. In order to understand the internal motion of telescope, the goal of this thesis is to build a state space model of a two-degree-of-freedom telescope using a known star location as feedback. The model presented in this thesis may be used for the telescope simulation and the control design in the future. This project considers a light weight flexible structured telescope and uses state space formulation to model dynamic motion of the telescope since state space model is better for modeling and analyzing a multiple-inputs and multiple-outputs system. A one-degree-freedom experimental test stand is built to generate a series of telescope motion data in order to examine whether the telescope simulation model can represent a real telescope system. To record the telescope motion, a wheel encoder mounted on the azimuth motor and a LED light density sensor installed at the end of telescope body are used as feedbacks. The results show that the state space model can simulate many system dynamic behaviors as shown in the experimental result; however, it is not feasible to obtain all of telescope’s physical parameters from the system feedbacks.
# Table of Contents

List of Figures ................................................................. vii  
List of Tables ........................................................................ viii  
List of Equations ............................................................... ix  

## Chapter 1: Introduction ................................................. 1

  1.1: Background ............................................................. 1  
  1.2: Star Tracking and Telescope Control ............................ 4  
    1.2.1 Altazimuth Tracking ...................................................... 4  
    1.2.2: Open Loop and Closed Loop Position Control .............. 6  
    1.2.3: Disturbances ................................................................. 8  
    1.2.4: Stability Criteria ......................................................... 10  
  1.3: Project Overview and Objective .................................... 11  

## Chapter 2: Model of Flexible Telescope .............................. 13

  2.1: Telescope System ......................................................... 13  
    2.1.1: Structure Components ................................................. 14  
    2.1.2: Control Inputs .............................................................. 16  
    2.1.3: Sensor Feedback ........................................................ 16  
  2.2: Model of External Disturbances .................................... 17  
    2.2.1: Gravity ..................................................................... 17  
    2.2.2: Wind .......................................................................... 18  
  2.3: Azimuth System ............................................................ 19  
    2.3.1: Model of Azimuth System ............................................ 19  
    2.3.2: Azimuth Model State Equations .................................. 22  
    2.3.3: State-space Model of Azimuth System ...................... 26  
  2.4: Altitude System ............................................................ 31  
    2.4.1: Model of Altitude System ............................................ 31  
    2.4.2: Altitude Model State Equations .................................. 34  
    2.4.3: State-space Model of Altitude System ...................... 39  
  2.5: Matlab Simulation ....................................................... 44  
    2.5.1: Simulink Model .......................................................... 44  
    2.5.2: Simulation Output ..................................................... 48  

## Chapter 3: Azimuth Experimental Telescope ......................... 52
### Chapter 3: Experiment Set Up

3.1: Experiment Set Up .......................................................................................... 5 2
   3.1.1: Telescope Structure .................................................................................. 5 2
   3.1.2: Software and Data Acquisition Interface .................................................. 5 4
   3.1.3: Guide Star Feedback Simulation ............................................................... 5 5

3.2: Experiments Result ......................................................................................... 5 8
   3.2.1: Light Feedback .......................................................................................... 5 9
   3.2.2: PI Control ................................................................................................. 6 1
   3.2.3: Tap Test .................................................................................................... 6 2
   3.2.4: Feedback Control Using Light Feedback ................................................. 6 5

### Chapter 4: Simulation Result Analysis ............................................................ 6 9

4.1: Turned Simulation model ............................................................................... 6 9
   4.1.1: PI Control Tuning ..................................................................................... 6 9
   4.1.2: Frequency Response of Simulation Model ................................................ 7 0

4.2: Controllability and Observability Analysis .................................................... 7 2

### Chapter 5: Conclusion ..................................................................................... 7 4

5.1: Light Weight Flexible Telescope Model .......................................................... 7 4
5.2: Experimental Test Stand .................................................................................. 7 4
5.3: Control Analysis ............................................................................................... 7 5
5.4: Current Project State and Future Work ........................................................... 7 6

### Reference .......................................................................................................... 7 7

Appendix A: Derivation of Azimuth Model .............................................................. 7 8
Appendix B: Disturbance Model .............................................................................. 8 6
Appendix C: Derivation of Altitude Model ............................................................... 8 9
Appendix D: Matlab Simulation Model .................................................................... 9 9
Appendix E: Matlab Plot Functions ....................................................................... 1 7 2
Appendix F: OTA Calculation ............................................................................... 1 7 6
Appendix G: System Parameters ........................................................................... 1 7 7
List of Figures

Figure 1: Stationary Telescope - Hooker telescope at Mount Wilson Observatory ... 2
Figure 2: Homemade telescope ................................................................. 3
Figure 3: Altitude (h) and Azimuth (A) coordination .................................. 5
Figure 4: Telescope Open Loop Control flow chart ................................. 7
Figure 5: Telescope Closed Loop Control flow chart ............................... 8
Figure 6: Generalized Telescope System .................................................. 13
Figure 7: Telescope System Components ................................................ 15
Figure 8: Azimuth Telescope Model ........................................................ 22
Figure 9: Azimuth Telescope Model Tree Diagram ................................... 23
Figure 10: Star Image Displacement ......................................................... 28
Figure 11: Altitude Telescope Model ....................................................... 34
Figure 12: Altitude Telescope Model Tree Diagram .................................. 36
Figure 13: State-space subsystem ............................................................. 45
Figure 14: Wind model for altitude system .............................................. 46
Figure 15: Gravity model ......................................................................... 46
Figure 16: Full telescope in Simulink ....................................................... 47
Figure 17: PID controller .......................................................................... 48
Figure 18: Result of azimuth simulation .................................................. 49
Figure 19: Result of altitude Simulation .................................................. 50
Figure 20: pointing path and star image feedback .................................... 51
Figure 21: Experimental Setup of experimental telescope ....................... 53
Figure 22: Motomatic Station ................................................................. 54
Figure 23: SC-2075 DAQ Interface .......................................................... 55
Figure 24: Simulated Star Source ............................................................ 56
Figure 25: Star Image sensor used on Test Stand ..................................... 56
Figure 26: Star image feedback simulation sensor situation ..................... 57
Figure 27: Light response from Simulation ............................................. 58
Figure 28: Telescope test ......................................................................... 59
Figure 29: Star Image feedback reading ................................................... 60
Figure 30: Telescope control response (PID control) ............................... 62
Figure 31: OTA response to tap test ....................................................... 64
Figure 32: Frequency response of telescope to tap test ............................ 65
Figure 33: PI control using star image feedback ..................................... 67
Figure 34: Turned Simulation Result ....................................................... 70
Figure 35: Frequency response of simulation model ............................... 71
List of Tables

Table 1: Telescope system components ............................................................ 15
Table 2: Control Inputs of telescope ................................................................. 16
Table 3: Sensor Feedback of telescope ............................................................. 17
Table 4: Modeling Parameters for Azimuth Model ........................................... 21
Table 5: Modeling parameters for Altitude Model (Part 1) ............................. 33
Table 6: Modeling parameters for Altitude Model (Part 2) ............................. 34
Table 7: Natural frequency of components ...................................................... 72
Table 8: Controllability and Observability Analysis ......................................... 72
List of Equations

Equation 1: Gravitational disturbance ............................................................... 18
Equation 2: Wind disturbance ........................................................................ 18
Equation 3: States of azimuth system ............................................................ 23
Equation 4: State space equation of azimuth model ......................................... 24
Equation 5: Torque transformation of azimuth model ....................................... 24
Equation 6: Transformed azimuth model States ................................................. 25
Equation 7: Transformed state equation for azimuth model.............................. 26
Equation 8: State Space Model for Azimuth .................................................... 27
Equation 9: Azimuth Model Input .................................................................. 27
Equation 10: Azimuth Model Output .............................................................. 27
Equation 11: Star Image Displacement Calculation .......................................... 28
Equation 12: Reduced Star Image Displacement Equation .............................. 28
Equation 13: Azimuth Model disturbance Source ........................................... 28
Equation 14: State (A) matrix for azimuth model ........................................... 29
Equation 15: Input (B) matrix for azimuth model ............................................ 29
Equation 16: Output (C) matrix for azimuth model ......................................... 30
Equation 17: Direct transmittance (D) matrix for azimuth model ................. 30
Equation 18: Disturbance (E) matrix for azimuth system ................................. 31
Equation 19: States of altitude system ........................................................... 35
Equation 20: State equation of altitude system ............................................... 37
Equation 21: Torque transformation of altitude model ................................. 38
Equation 22: Altitude Model State ................................................................. 38
Equation 23: Transform state equations for altitude model ............................. 39
Equation 24: State Space Model for Altitude ................................................... 40
Equation 25: Altitude Model Input ................................................................. 40
Equation 26: Altitude Model Output ............................................................. 40
Equation 27: Star Image Displacement Calculation for altitude model ........... 41
Equation 28: Reduced Star Image Displacement Equation of altitude model .. 41
Equation 29: Altitude Model Disturbance Source .......................................... 41
Equation 30: System (A_a) matrix for altitude model ..................................... 41
Equation 31: Input (B_a) matrix for altitude model ........................................ 42
Equation 32: Output (C_a) matrix for altitude model ...................................... 43
Equation 33: Direct transmittance (D_a) matrix for altitude model ............... 43
Equation 34: Disturbance (E_a) matrix for altitude model ............................. 44
Chapter 1: Introduction

1.1: Background

The accuracy of optical telescopes in the past time was dominated by the construction of the telescope body and the precision of optical surface. In order to have an accurate optical telescope, the telescope must have an extremely rigid and heavy structure. The rigid telescope body can decrease structure deformation under external disturbances. A heavy telescope structure can help reduce vibration caused by impulse disturbance and telescope tracking motion. This kind of telescope (shown in Figure 1) is stationary and constructed inside a shelter which makes it free from many environment disturbances. Moreover, the gear ratio in the drive chain is big in order to support and control heavy telescope structure; therefore the tracking motion is very slow.
The FEA (Finite Element Analysis) model is a popular method for structure modeling for big telescopes. However, it is hard to design a control system using FEA model generated by a computer due to its complicated and usually oversized model. This kind of telescope control heavily relies upon empirical data. Each telescope is tested. Data Results are used to create a LUT (Lookup Table). The LUT includes the amount deformation occurs on the telescope structure under external disturbances. This LUT is be used as reference for control calibration.[1]

Some other telescope, because of its need to move and operate at different location, is constructed with a much lighter structure. An amateur

---

telescope is an example of it (Shown in Figure 2). This kind telescope is required to be disassembled and moved to a remote location, usually outdoors. The operator will put it together at the location. The light structure telescope is susceptible to structure vibration and external disturbances. The first reason is that the maximum weight of the telescope is limited by the transportation tool and human power that put it together. Due to this weight limit, the construction of telescope isn’t as stiff and heavy as stationary telescope. Second, a light structure telescope is usually operated in an environment with more external disturbances. Since there is no shelter or dome to protect it, disturbances like wind or a wild animal can directly affect the telescope.

![Figure 2: Homemade telescope](image)

Besides being susceptible to vibration and deformation, it is also hard to calibrate a light structure telescope. Even though the telescope calibration

---

strategy using empirical data is common for a big stationary telescope, the same method is less useful for the light structure telescope. This is because of the uncontrollable environment and errors in the set up process. The telescope structure will be slightly different every single time. The calibration process is also time consuming and requires many precise measuring tools. It isn't feasible to perform it on a low price telescope every time an operator sets it up. On the other hand, modern telescopes have additional sensors. Many of these sensors could provide extra information about the telescope. Moreover, a high torque motor could be used as a direct drive motor on a light structure. The drive chain could be eliminated since the inertia of such a telescope is low. It offers a faster response and a more linear system. With the faster computer used today, it is possible to apply active control with additional feedback to improve performance of a light structure telescope.

1.2: Star Tracking and Telescope Control

1.2.1 Altazimuth Tracking

Many light structure telescopes are manual operated telescope. It requires the operator to manually point it at the target star. Since the star actually travels across the sky over night, the operator also needs to adjust its pointing so the target star will stay in the field of view. Some telescopes are computer controlled so it can track a single star over the sky in a long period of time. Two main
tracking strategies are used today.

1. Altazimuth Mounts

2. Equatorial Fork Mounts

Altazimuth method uses two motors with position control to control the azimuth and altitude of the telescope. The altitude and azimuth system is the coordination used to describe the pointing direction of telescope. Altitude and azimuth coordination is shown in Figure 3. In this figure, “h” represents altitude angular position. Altitude is measured as the angle from horizontal to the direction which the telescope pointing and it goes from 0 degrees to 180 degrees. ”A” represents the azimuth angular position. Azimuth is measured as the angle from north of the direction of the telescope is pointing from a horizontal plane and it goes from 0 degrees to 360 degrees. These two values can fully define the telescope point in a half-sphere above the horizon.

Figure 3: Altitude (h) and Azimuth (A) coordination
Altazimuth tracking is used for the telescope system discussed in this project since it is used by most of research telescopes and amateur telescopes today. Altazimuth tracking is more mechanically complicated compared to equatorial tracking. The equatorial tracking has a rotational axis parallel to the Earth’s axis of rotation; hence it only needs to rotate about one single axis at a constant speed to follow a star in the sky. The Altazimuth tracking requires two axes, azimuth and altitude, to track a star over the sky. Two axes have to continually move at a different rate in order to track a celestial target. However, equatorial tracking requires a massive and precise mounting structure in order to make sure it is parallel to the Earth’s axis of rotation. For an amateur, it is easier and cheaper to construct a two axes system than design a heavier and more expensive equatorial mount. The complex control issue can usually be solved by the integration of a microcontroller or computer controlled system.

1.2.2: Open Loop and Closed Loop Position Control

The open loop position control indicates that the telescope doesn’t know where the star is [2]. It only knows which direction it is pointing. The open loop position control uses position sensors on the telescope rotation axis to measure the pointing angle. In the case of the Altazimuth mount, there is one encoder on each axis (Shown in Figure 4). These sensors are used to measure drive motor angular displacement. Then control system can calculate the altitude angle and azimuth angle of the telescope. Most of light structure telescope uses this
method to track a star.

![Figure 4: Telescope Open Loop Control flow chart](image)

However, a telescope using an open loop control doesn't know if it is actually pointing at the target star. The vibration and deformation in the OTA (Optical Telescope Assembly) affects telescope pointing. They create errors that are not observable by the position sensors placed on the end of a drive chain. Though position sensors suggest that the telescope is pointing at the correct direction; the combined angular position, telescope pointing direction plus deformation of OTA, might not be correct. Most of stationary telescopes use empirical data to calibrate the control system. This method adds or subtracts a certain amount of an angle when the telescope is at a different location. For example: if the telescope is trying to aim at a star at a 24 degrees altitude, the LUT fixes it to 26 degrees. Two extra degrees is compensated for telescope deformation due to gravity. This method is also not feasible to a light structure telescope due to cost of time and precise tools. It also does not help compensate for external disturbance that is not repeatable.

Close loop position control uses the star as a feedback signal. The star sensor forms a second feedback loop (Shown in Figure 5). This method uses a star image sensor to measure the displacement of a reference star (a
bright, not-interfered star at known location is usually picked as reference star because it gives clearer location feedback) from where it is supposed to be. This information can be used for fine adjustment of telescope pointing. Since it is using an observed target as its feedback source, this method makes sure that the combined angular position is correct.

![Telescope Closed Loop Control flow chart]

**Figure 5: Telescope Closed Loop Control flow chart**

### 1.2.3: Disturbances

Gravity produces repeatable deformation [2]. The effect of gravity on the telescope is due to the deformation of the telescope structure. The deformation is always in the vertical direction. OTA deformation due to gravitation is a function of the altitude angular position. If the altitude is small, the telescope will have a bigger deformation. Gravitational disturbances only exist in an altitude model. It won't affect the telescope motion on a horizontal plane.

Wind produces non-repeatable deformation [2]. The disturbance caused by wind has a slow and fast component. The slow component of wind disturbance is the steady wind. It applies a long and steady force on the telescope structure and causes the telescope to drift away from its target.
The steady wind is usually large in amplitude and could cause large telescope deformation. However, behavior of steady wind is usually more predictable. The fast component of the wind disturbance is wind gust. The wind gust is characterized by an abrupt and nearly discontinuous force on telescope body. However, the amplitude of wind gust is smaller and doesn't create as much deformation as a steady wind. The gust is usually 20% to 10% of steady wind [3]. The steady wind information can be monitored by an external weather station or observed using a feedback star image. The wind gust acts like noise from the environment. Due to gust’s short time constant, it is hard to observe it from a weather station and is usually observed through a star feedback system.

The telescope model models the wind as a torque applied on the motor shaft. The amplitude of the torque caused by steady wind is a function of wind disturbance amplitude, wind direction, and the position of telescope at the moment. If the telescope structure is perpendicular to the wind direction, the effect of the wind is the biggest. The steady wind source is modeled by a square wave. The wind gust is white noise with amplitude equal to 20% to 10% of the square wave [3].

Like the wind, temperature also produces non-repeatable deformation. The change of temperature which affects telescope accuracy occurs on a rather long time; this is because the sudden change of air temperature usually has small amplitude and the telescope component usually has a large thermal time constant; therefore, the small temperature change usually won’t change telescope geometry. The large temperature change which affects
telescope geometry occurs on a time scale of hours and days. Since the project focuses on the transient response, the telescope model does not count for the temperature in the environment.

Other disturbances include human contact, atmosphere air density change, ground vibration, and other light pollution. The accidental touching of the telescope structure can be modeled as a stronger wind gust. Other effects generally have less amplitude comparing with major disturbance and are ignored in the current model.

1.2.4: Stability Criteria

The response of telescope to control input and environmental disturbances changes the direction of telescope pointing. For a stable telescope, this change of pointing needs to remain in tolerance over a certain length of time. If the change of the angular pointing direction is too big, the star image will become blurring. The tolerance is determined by the observation target. The change of the angular pointing direction should not change by more than 1/10 of the beam’s width. The equation is $\Delta \theta \leq \theta / 10 \approx (\lambda / D) / 10$ [2]. $\lambda$ is wavelength of observation target and D is the main reflector diameter. The pointing stability is determined by the structure of the telescope.
1.3: Project Overview and Objective

Modern telescope is an integrated mechanical and electrical system. Its accuracy is challenged by telescope structure and external disturbances, as gravity, wind, and temperature change. Another issue is the internal friction in the telescope system. The internal friction is an issue for a light weight flexible telescope because this kind of telescope is not as well constructed as a VLT (very large telescope). It takes the telescope extra effort to overcome the static friction when it starts moving and usually causes an overshoot. It is hard to design a high performance control system for a light weight flexible telescope. Its light structure makes it susceptible to vibration and deformation. It is easily disturbed and usually operated under many environmental disturbances. Most importantly, many techniques used to calibrate research telescope can’t be used on an individual telescope. In order to design a control system, it is important to gain a better understanding of the telescope system. Many of light structure telescopes are custom-build and modified for the user. Each of them has distinct structure characteristics, and therefore a specific model does not help too much.

This project describes a light structure flexible telescope as a generalized flexible telescope structure. It fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. The model of a flexible telescope provides a telescope dynamic in state-space format since state space model is better for modeling and analyzing a multiple inputs and multiple outputs system. The model includes a transient flexible
response in order to represent the telescope response to a control input or a disturbance. It models the telescope tracking motion but also the vibration of each vibrating elements. In this case, it is important that the model can demonstrate the first vibration mode and simulates pointing error of the telescope since the first vibration mode causes the greatest pointing error of the telescope tracking. This model also models environment disturbance. Gravity and wind is considered in the model. The model concentrates on modeling the transient response of telescope. The effect of temperature change is not considered since its time constant is a lot longer.

The goal of this project is to construct a model for a light weight flexible telescope serving as a platform for testing active control system and system identification. In order to exam that if the telescope simulation model can represent a real telescope system, a one degree freedom experimental test stand is built to generate a series of telescope motion data. The simulation model is not a dynamic model for this experimental test stand; however, the model should be able to reproduce many system patterns shown in the experimental data since the general shape of simulation model and experimental is the same. The simulation of an experimental test stand is analyzed to determine the controllability and observability of the system. It shows the minimum requirement of sensor feedbacks in order to enable the operator to capture all the major and minor motion during the telescope motion.
Chapter 2: Model of Flexible Telescope

This chapter presents the modeling of the light weight flexible telescope mechanical and electrical system. The telescope structure consists of azimuth motor, altitude motor, telescope fork, OTA (Optical Telescope Assembly), mirror(main reflector), and mirror mount. The telescope electronics includes two driving motors. This model also counts control inputs used to perform telescope tracking and possible external disturbances that affect telescope performance.

2.1: Telescope System

To model the motion of a flexible telescope, a general shape
telescope with Altazimuth tracking system is considered (shown in Figure 6). The cylindrical body on the top is the body of telescope. It contains the OTA, the main reflector (mirror), and the mirror mount. The bottom part is the telescope mount. It contains the fork, two driving motor, and all the controller components. The generalized telescope model, without flexibility, is a system with two degree of freedom. The telescope rotates around two axes. Azimuth motion rotates around Z-axis and altitude motion rotates around X-axis. Two axes intersect at center of gravity of telescope body; therefore, without disturbance, the telescope should be a stable system which can stay still at any position.

2.1.1: Structure Components

The generalized telescope mechanical system has five major components (Shown in Figure 7 and Table 1). Red dot on Figure 7 is the center of gravity and the rotational center of the telescope body. Mirror mount is the rear end of the telescope body which holds telescope mirror. OTA extends out from the mirror mount. The observing point is at the opposite end of the OTA, illustrated with a green dot on the Figure 7. The fork is the entire supporting mount of the telescope. When models a telescope mechanical system, two motor and drive chain is part of the fork. The fork, OTA and the mirror mount together is the telescope structure.
Two motors are the electrical-mechanical system. They convert electric current in the electrical system into torque. Azimuth motor and altitude motor provide torque to two axes of telescope mechanical system. The generalized telescope uses direct drive motor; therefore, there is no gear chain between telescope axes and motor. They change angle between telescope body and telescope mount. These two motor receive control input from the position controller.

<table>
<thead>
<tr>
<th>Parts Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth Motor</td>
<td>Motor that controls azimuth motion of telescope</td>
</tr>
<tr>
<td>Altitude Motor</td>
<td>Motor that controls altitude motion of telescope</td>
</tr>
<tr>
<td>Fork</td>
<td>The stand of telescope; support OTA and lift it from the ground</td>
</tr>
<tr>
<td>OTA</td>
<td>Front part of telescope body, from where fork contacts with body to where image receiver is</td>
</tr>
<tr>
<td>Mirror and Mirror Mount</td>
<td>Rear part of telescope body, from where fork contacts with body to where primary mirror is</td>
</tr>
</tbody>
</table>
2.1.2: Control Inputs

Control inputs are part of the electric system of the telescope. The generalized telescope control system has two inputs (shown in Table 2). Telescope control inputs are generated by the position controller. These two voltage signals are sent to the power amplifier. The amplified signals are sent to the motor and used to control motor output.

Table 2: Control Inputs of telescope

<table>
<thead>
<tr>
<th>Input Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth Motor</td>
<td>Voltage input, controls azimuth motion</td>
</tr>
<tr>
<td>Altitude Motor</td>
<td>Voltage input, controls altitude motion</td>
</tr>
</tbody>
</table>

2.1.3: Sensor Feedback

Telescope systems have three different sensor feedbacks (shown in Table 3). First two sensors are azimuth encoder and altitude encoder. In a generalized telescope model, they are considered as ideal position feedback sensors which give motor angular position in radius with an infinite small resolution. They are directly connected to the motor. Since there is not gear chain in the mechanical system, the outputs are also an angular position of general telescope pointing direction. The feedbacks from encoders are illustrated by an arrow between position sensor and position controller.

The star image sensor is mounted on the extended end of OTA. Its location is illustrated with the green dot in Figure 7. In the generalized telescope model, the output of star image sensor is the displacement between desired
telescope pointing center and combined pointing center, which also counts for the fork, OTA, and mirror mount vibration. It produces a 2 degrees displacement on XY coordination. The feedback from the star image sensor is illustrated by arrow between the reflectors/receivers and the position controller. In the generalized model, the sensor images a reference star that is always at the actual center of telescope pointing; therefore, always provides a feedback. In a real system, the star image feedback only works when the reference star is in the field of view, which means the telescope open loop controller has to be able to bring the telescope pointing direction close to the reference star location before the star image sensor starts to work.

Table 3: Sensor Feedback of telescope

<table>
<thead>
<tr>
<th>Feedback Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth encoder</td>
<td>Gives angular position and angular velocity in azimuth direction</td>
</tr>
<tr>
<td>Altitude encoder</td>
<td>Gives angular position and angular velocity in altitude</td>
</tr>
<tr>
<td>Star image</td>
<td>Gives target star location in XY coordination</td>
</tr>
</tbody>
</table>

2.2: Model of External Disturbances

2.2.1: Gravity

The gravitational disturbance is modeled as torque applied on the altitude model. Gravitational torque is a function of telescope length, telescope pose, and the mass of instrument mounted on the end of upper OTA and mirror mount. The calculation is shown in Equation 1.
**Equation 1: Gravitational disturbance**

\[
T_{\text{Gravity}} = -m_{\text{instrument}} \times g \times \cos(\theta_{\text{axis}}) \times L_{\text{telescope}}
\]

The angular position change of the telescope in the altitude system is from 0 to 90 degrees; therefore, the torque is nonlinear. The torque is calculated externally from the system model and brought in the model as input. Gravitational disturbances caused by the OTA and the mirror mount should balance each other since the telescope is a balanced system. However, gravitational force still bends the OTA and the mirror mount slightly hence create a star image displacement.

**2.2.2: Wind**

The wind disturbance is modeled as torque applied on both the altitude and azimuth model. The pressure from the wind force is \( P_{\text{wind}} = 0.5 \rho_{\text{air}} V_w^2 C \) [6]. \( V_w \) is the speed of wind and \( C \) is the shape factor of the telescope body. \( C \) is assumed to be 1 in this project since the true shape of telescope is unknown. From wind pressure, the torque applied on rotational center of telescope is calculated by multiplying the wind pressure by the telescope section area and half of the length of the telescope (shown in Equation 2).

**Equation 2: Wind disturbance**

\[
T_{\text{wind}} = 0.5 \times \rho_{\text{air}} \times [V_{\text{wind}} \times \cos(\theta_j - \theta_{\text{wind}}) \times \sin \theta_j]^2 \times A_{\text{telescope}} \times 0.5 \times L_{\text{telescope}}
\]

The angular position change of the telescope is 0 to 90 degree in the
altitude system and 0 to 360 degree in the azimuth system; therefore, the torque is nonlinear. The torque is calculated externally from the system model and brought in the model as input.

2.3: Azimuth System

Generalized telescope model is separated into two parts: azimuth model and altitude model. The azimuth and the altitude system is treated as uncoupled in current model; however, in a real telescope there are some coupling between those axes. One of the coupling sources is the gyroscopic effect. Nevertheless, the telescope is usually operated under low speed with very low rotational angular acceleration; therefore, coupling between two subsystems can be ignored. The azimuth model describes telescope motion around Z axis, and the altitude model describes telescope motion around X axis.

2.3.1: Model of Azimuth System

Structure components of azimuth model include azimuth motor, OTA, mirror mount, and Fork. In the azimuth model, the altitude model is considered part of the fork. The relationship between each structure component is shown in Figure 8. The springs and dampers shown in the figure are torsional springs and torsional dampers. The azimuth motor is connected to the ground which provides absolute zero reference for the angular velocity of mechanical system. The azimuth system has one control input. When input voltage is sent to azimuth
motor, rotor A will gain velocity while velocity of rotor B will be zero. The angle between rotor A and rotor B is $\Theta_j$. A rotational spring and damper connects rotor A and the fork/altitude motor. The angular position of the fork is $\Theta_{Jam}$. This angular position is the absolute angular position, the angle between the fork and ground zero (North). The fork/altitude is connected to the mirror mount and OTA with two separated rotational springs and damper sets. When the fork/altitude motor displaces, torque will be sent to the mirror mount and OTA through the spring and damper set; therefore, moving mirror mount and OTA. The entire system will move with the input from the azimuth motor. The angular position of OTA is $\Theta_O$ and angular position of mirror mount is $\Theta_m$. The detail parameters for azimuth model are shown in Table 4.)
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J$</td>
<td>Inertia of azimuth control motor</td>
<td>$\text{kg} \cdot \text{m}^2$</td>
</tr>
<tr>
<td>$J_O$</td>
<td>Inertia of OTA</td>
<td>$\text{kg} \cdot \text{m}^2$</td>
</tr>
<tr>
<td>$J_m$</td>
<td>Inertia of mirror mount</td>
<td>$\text{kg} \cdot \text{m}^2$</td>
</tr>
<tr>
<td>$J_f$</td>
<td>Inertia of fork and altitude motor</td>
<td>$\text{kg} \cdot \text{m}^2$</td>
</tr>
<tr>
<td>$K_O$</td>
<td>Spring of OTA around z axis in horizontal direction</td>
<td>$\text{N} \cdot \text{m} / \text{radius}$</td>
</tr>
<tr>
<td>$K_m$</td>
<td>Spring of mirror mount around z axis in horizontal direction</td>
<td>$\text{N} \cdot \text{m} / \text{radius}$</td>
</tr>
<tr>
<td>$K_f$</td>
<td>Spring of fork around z axis in horizontal direction</td>
<td>$\text{N} \cdot \text{m} / \text{radius}$</td>
</tr>
<tr>
<td>$B$</td>
<td>Damping coefficient of azimuth control motor</td>
<td>$\text{N} \cdot \text{m} \cdot \text{s} / \text{radius}$</td>
</tr>
<tr>
<td>$B_O$</td>
<td>Damping coefficient of OTA in horizontal direction</td>
<td>$\text{N} \cdot \text{m} \cdot \text{s} / \text{radius}$</td>
</tr>
<tr>
<td>$B_m$</td>
<td>Damping coefficient of mirror mount in horizontal direction</td>
<td>$\text{N} \cdot \text{m} \cdot \text{s} / \text{radius}$</td>
</tr>
<tr>
<td>$B_f$</td>
<td>Damping coefficient of fork around z axis</td>
<td>$\text{N} \cdot \text{m} \cdot \text{s} / \text{radius}$</td>
</tr>
<tr>
<td>$R$</td>
<td>Internal resistance of azimuth motor system</td>
<td>$\Omega$</td>
</tr>
<tr>
<td>$L$</td>
<td>Internal inductance of azimuth motor system</td>
<td>$H$</td>
</tr>
<tr>
<td>$K_a$</td>
<td>Motor constant of azimuth motor system</td>
<td>$\text{N} \cdot \text{m} / \text{Amps}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>States</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_J$</td>
<td>Angular position of azimuth pointing</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\theta_J O$</td>
<td>Angular position of OTA</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\theta_J m$</td>
<td>Angular position of mirror mount</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\theta_J f$</td>
<td>Rotation angle of fork and altitude motor around z axis</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\Omega_J$</td>
<td>Azimuth angular velocity of azimuth pointing</td>
<td>$\text{Radius} / \text{s}$</td>
</tr>
<tr>
<td>$\Omega_J O$</td>
<td>Azimuth angular velocity of OTA</td>
<td>$\text{Radius} / \text{s}$</td>
</tr>
<tr>
<td>$\Omega_J m$</td>
<td>Azimuth angular velocity of mirror mount</td>
<td>$\text{Radius} / \text{s}$</td>
</tr>
<tr>
<td>$\Omega_J f$</td>
<td>Azimuth angular velocity of fork and altitude motor</td>
<td>$\text{Radius} / \text{s}$</td>
</tr>
<tr>
<td>$i_L$</td>
<td>Current through azimuth control motor</td>
<td>$\text{Amps}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_s$</td>
<td>Input voltage for azimuth control motor</td>
<td>$\text{V}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{s1}$</td>
<td>Wind disturbance act on OTA for azimuth model</td>
<td>$\text{N} \cdot \text{m}$</td>
</tr>
<tr>
<td>$T_{s2}$</td>
<td>Wind disturbance act on mirror mount for azimuth model</td>
<td>$\text{N} \cdot \text{m}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_J$</td>
<td>Angular position of azimuth as output</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\Omega_J$</td>
<td>Azimuth angular velocity</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$X_g$</td>
<td>Guide star feedback sensor for azimuth coordination</td>
<td>$\text{Radius}$</td>
</tr>
</tbody>
</table>
2.3.2: Azimuth Model State Equations

Graphical technique is used to represent the dynamics system model [5]. The linear graph represents the topological relationships of lumped-element interconnections within system [5]. Linear graph of azimuth model is shown in Figure 9. This graph is used represents both the electrical and mechanical system. The loop on the left is the model of DC motor. The bigger loop on the right is the model of the telescope mechanical system. The component connecting the two systems is electric mechanical component motor. The motor serve as a transformer in between two loops transfer current (shown as 1 on Figure 9) to torque (shown as 2 on Figure 9). From the linear graph, the system has 8 states (Shown in Equation 3).
Equation 3: States of azimuth system

\[ X = \begin{bmatrix} \Omega_j, \Omega_{jam}, \Omega_{jo}, \Omega_{jm}, T_{Kr}, T_{Ko}, T_{Km}, I_L \end{bmatrix} \]

Figure 9: Azimuth Telescope Model Tree Diagram

Eight state variables are used to generate 8 state equations to represent dynamic motion in the azimuth system (Shown in Equation 4). The detail derivation is shown in Appendix A.
Equation 4: State space equation of azimuth model

\[
\frac{d\Omega_j}{dt} = \frac{1}{J} \left[ \frac{1}{K_a} i_L - B\Omega_j - T_{Kf} - B_f (\Omega_j - \Omega_{Jam}) \right]
\]

\[
\frac{d\Omega_{Jam}}{dt} = \frac{1}{J_{am}} \left[ T_{Kf} + B_f (\Omega_j - \Omega_{Jam}) - T_{Ko} - B_o (\Omega_{Jam} - \Omega_{Jo}) - T_{Km} - B_m (\Omega_{Jam} - \Omega_{Jm}) \right]
\]

\[
\frac{d\Omega_{Jo}}{dt} = \frac{1}{J_o} \left[ T_{Ko} + B_o (\Omega_{Jam} - \Omega_{Jo}) + T_{s1} \right]
\]

\[
\frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} \left[ T_{Km} + B_m (\Omega_{Jam} - \Omega_{Jm}) + T_{s2} \right]
\]

\[
\frac{di_L}{dt} = \frac{1}{L} \left( V_s - Ri_L - \frac{1}{Ka} \Omega_j \right)
\]

\[
\frac{dT_{Kf}}{dt} = \frac{1}{J_{am}} (\Omega_j - \Omega_{Jam})
\]

\[
\frac{dT_{Ko}}{dt} = \frac{1}{J_o} (\Omega_{Jam} - \Omega_{Jo})
\]

\[
\frac{dT_{Km}}{dt} = \frac{1}{J_m} (\Omega_{Jam} - \Omega_{Jm})
\]

Four states \( T_{Kf}, T_{Ko}, T_{Km} \) shown in Equation 3 are hard to be measured in the azimuth system with accessible sensors. However, it is easier to transform them into a different format. Torque between two components can be rewritten as angular displacement multiplied by the rotational spring constant (Shown in Equation 5).

Equation 5: Torque transformation of azimuth model

\[
T_{Kf} = K_f (\theta_j - \theta_{Jam})
\]

\[
T_{Ko} = K_o (\theta_{Jam} - \theta_O)
\]

\[
T_{Km} = K_m (\theta_{Jam} - \theta_m)
\]

Azimuth model states have now become angular positions and angular
velocities of system components and current in the electric system. New azimuth models states is shown in Equation 6. With the new state variables, state equation is transformed also. New state equation is shown in Equation 7.

**Equation 6: Transformed azimuth model States**

\[
X = \begin{bmatrix}
\theta_j \\
\theta_{jm} \\
\theta_{j_o} \\
\theta_{jm} \\
\Omega_j \\
\Omega_{j_f} \\
\Omega_{j_o} \\
\Omega_{jm} \\
i_L
\end{bmatrix}_{8 \times 1}
\]
**Equation 7: Transformed state equation for azimuth model**

\[
\frac{d\theta_j}{dt} = \Omega_j \\
\frac{d\theta_{jam}}{dt} = \Omega_{jam} \\
\frac{d\theta_{jo}}{dt} = \Omega_{jo} \\
\frac{d\theta_{jm}}{dt} = \Omega_{jm} \\
\frac{d\Omega_j}{dt} = \frac{1}{J} \left[ \frac{1}{K_a} i_L - B \Omega_j - K_f (\theta_j - \theta_{jam}) - B_f (\Omega_j - \Omega_{jam}) \right] \\
\frac{d\Omega_{jam}}{dt} = \frac{1}{J_{jam}} \left[ K_f (\theta_j - \theta_{jam}) + B_f (\Omega_j - \Omega_{jam}) - K_a (\theta_{jam} - \theta_{jo}) - B_a (\Omega_{jam} - \Omega_{jo}) - K_m (\theta_{jam} - \theta_{jm}) - B_m (\Omega_{jam} - \Omega_{jm}) \right] \\
\frac{d\Omega_{jo}}{dt} = \frac{1}{J_{jo}} \left[ K_f (\theta_j - \theta_{jam}) + B_f (\Omega_j - \Omega_{jam}) - K_a (\theta_{jam} - \theta_{jo}) - B_a (\Omega_{jam} - \Omega_{jo}) + T_{s1} \right] \\
\frac{d\Omega_{jm}}{dt} = \frac{1}{J_{jm}} \left[ K_m (\theta_{jam} - \theta_{jm}) + B_m (\Omega_{jam} - \Omega_{jm}) + T_{s2} \right] \\
\frac{di_L}{dt} = \frac{1}{L} \left( V_s - Ri_L - \frac{1}{K_a} \Omega_j \right)
\]

**2.3.3: State-space Model of Azimuth System**

8 states variables are the minimum set of variables which can be used to describe azimuth system. System outputs are system variables that are accessible and can be measured by the sensors (shown in Equation 10). There are 3 system outputs we can obtain from sensors. They can generate 3 output equations. 8 state variables are substituted with 9 state variables. These 9 state equations and output equations are rewritten in state-space format (shown in Equation 8).
Equation 8: State Space Model for Azimuth
\[
\dot{X} = A_{9x9}X_{9x1} + B_{9x1}U_{1x1} + E_{9x2}d_{2x1} \\
\dot{Y} = C_{3x9}X_{9x1} + D_{3x1}U_{1x1}
\]

The system input of azimuth model is voltage sent to the electric system
(Shown in Equation 9)

Equation 9: Azimuth Model Input
\[U = [V_i]_{1x1}\]

There are 3 system outputs. Angular position and angular velocity of the
motor could be obtained from azimuth encoder. \(X_t\) is the output from the star
image sensor. It is the displacement of the reference star.

Equation 10: Azimuth Model Output
\[
Y = \begin{bmatrix}
\theta_j \\
\Omega_j \\
X_t
\end{bmatrix}_{3x1}
\]

\(X_t\) is a function of mirror mount angular position and OTA angular position
of the azimuth system (shown in Figure 10). Angular displacements of the OTA
and the mirror mount both create image displacements. The star image
displacement is a combination of these two displacements. This relationship is
listed in Equation 11. Using small angle approximation, Equation 11 can be
linearized into a simpler format (shown in Equation 12). Notice the term \(\theta_{sym}\)
canceled in the equation. Vibration of fork does not affect azimuth pointing.
Figure 10: Star Image Displacement

Equation 11: Star Image Displacement Calculation
\[ X_i = L_{OTA} \times \sin\{2[(\theta_{jm} - \theta_{Jam}) + (\theta_{Jam} - \theta_j)]\} - L_{OTA} \times \sin[(\theta_{Jo} - \theta_{Jam}) + (\theta_{Jam} - \theta_j)] \]

Equation 12: Reduced Star Image Displacement Equation
\[ X_i = -L_{OTA}\theta_J - L_{OTA}\theta_{Jo} + 2L_{OTA}\theta_{jm} \]

There are two disturbance sources in the azimuth model. The wind torque applied on OTA and the wind torque applied on mirror mount. In azimuth motion, wind effect on fork is small; therefore, it is ignored.

Equation 13: Azimuth Model disturbance Source
\[ d = \begin{bmatrix} T_{x1} \\ T_{x2} \end{bmatrix} \]

All system matrices can be determined by transformer 9 state equations and 3 output equations. System matrices are shown below. A matrix shown in Equation 14 is the state matrix of the azimuth system.
Equation 14: State (A) matrix for azimuth model
\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
\frac{(K_O + K_m)}{J} & 0 & \frac{K_O}{J} & \frac{K_m}{J} & \frac{(B + B_O + B_m)}{J} & B & \frac{B_O}{J} & \frac{B_m}{J} & \frac{1}{K_a J} \\
0 & -\frac{K_f}{J_f} & 0 & 0 & \frac{B}{J_f} & -\frac{(B + B_f)}{J_f} & 0 & 0 & -\frac{1}{K_a J_f} \\
\frac{K_O}{J_O} & 0 & -\frac{K_O}{J_O} & 0 & \frac{B_O}{J_O} & 0 & -\frac{B_O}{J_O} & 0 & 0 \\
\frac{K_m}{J_m} & 0 & 0 & -\frac{K_m}{J_m} & \frac{B_m}{J_m} & 0 & 0 & -\frac{B_m}{J_m} & 0 \\
0 & 0 & 0 & 0 & -\frac{1}{K_a L} & \frac{1}{K_a L} & 0 & 0 & -\frac{R}{L} \\
\end{bmatrix}_{9 \times 9}
\]

B matrix shown in Equation 15 is the input matrix of the azimuth system.

Equation 15: Input (B) matrix for azimuth model
\[
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
1 \\
\frac{1}{L} \\
\end{bmatrix}_{9 \times 1}
\]

C matrix shown in Equation 16 is the output matrix of the azimuth system.
Output variables \( \theta_j \) and \( \Omega_j \) is a direction output from state variables. \( X \) is calculated using Equation 12.

**Equation 16: Output (C) matrix for azimuth model**

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

D matrix shown in Equation 17 is the direct transmittance matrix for the azimuth model. In the azimuth system, input of the system doesn't affect outputs; therefore transmittance matrix is a zero matrix.

**Equation 17: Direct transmittance (D) matrix for azimuth model**

\[
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
\]

E matrix shown in Equation 18 is the disturbance matrix for the azimuth system.
Equation 18: Disturbance (E) matrix for azimuth system

\[
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\frac{1}{J_o} & 0 \\
0 & \frac{1}{J_m} \\
0 & 0 \\
\end{bmatrix}^{9 \times 2}
\]

2.4: Altitude System

Altitude model describes the telescope motion around the X axis.

2.4.1: Model of Altitude System

The structure of the altitude model is similar to the azimuth model. Components include altitude motor, OTA, mirror mount, and Fork. In the altitude model, the azimuth model is considered part of the ground. The relationship between each structure component is shown in Figure 11. The springs and dampers shown in the figure are torsional springs and torsional dampers. In the altitude model, the fork is connected to the ground which provides absolute zero reference for the angular velocity of mechanical system. The altitude motor sits on the top of the fork and provide torque source to all 3 other components. The altitude system has one control input. When input voltage is sent to the altitude
motor, both rotor A, which is connected to mirror mount and OTA, and rotor B, which is connected to the fork will gain velocity. The angle between rotor A and rotor B of altitude motor is $\theta_{ja}$. Rotor B is connected to the fork and has the same angular displacement as fork $\Theta_f$. Since the fork is connected to the ground, the angular displacement of rotor B is limited by the spring element of fork. The altitude motor is connected to the mirror mount and OTA with two separated rotational spring and damper sets. When altitude motor displaces, torque will be sent to mirror mount and OTA through the spring and damper set; therefore, moving mirror mount and OTA. The OTA and mirror mount move with rotor A of the altitude motor. The angular position of fork oscillates and goes to zero. The angular position of OTA is $\theta_{o-a}$ and angular position of mirror mount is $\theta_{m-a}$. The detail parameters for azimuth model are shown in Table 5 and table 6.
<table>
<thead>
<tr>
<th>Altitude Model Parameters</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_{Ja} )</td>
<td>Inertia of altitude control motor</td>
<td>( kg \cdot m^2 )</td>
</tr>
<tr>
<td>( J_{Ja} )</td>
<td>Inertia of OTA</td>
<td>( kg \cdot m^2 )</td>
</tr>
<tr>
<td>( J_{Ja} )</td>
<td>Inertia of mirror mount</td>
<td>( kg \cdot m^2 )</td>
</tr>
<tr>
<td>( J_{Ja} )</td>
<td>Inertia of fork</td>
<td>( kg \cdot m^2 )</td>
</tr>
<tr>
<td>( K_{Ja} )</td>
<td>Spring of OTA around z axis in vertical direction</td>
<td>( N \cdot m / radius )</td>
</tr>
<tr>
<td>( K_{Ja} )</td>
<td>Spring of mirror mount around z axis in vertical direction</td>
<td>( N \cdot m / radius )</td>
</tr>
<tr>
<td>( K_{Ja} )</td>
<td>Spring of fork around z axis in vertical direction</td>
<td>( N \cdot m / radius )</td>
</tr>
<tr>
<td>( B_{Ja} )</td>
<td>Damping coefficient of altitude control motor</td>
<td>( N \cdot m \cdot s / radius )</td>
</tr>
<tr>
<td>( B_{Ja} )</td>
<td>Damping coefficient of OTA in vertical direction</td>
<td>( N \cdot m \cdot s / radius )</td>
</tr>
<tr>
<td>( B_{Ja} )</td>
<td>Damping coefficient of mirror mount in vertical direction</td>
<td>( N \cdot m \cdot s / radius )</td>
</tr>
<tr>
<td>( B_{Ja} )</td>
<td>Damping coefficient of fork around z axis</td>
<td>( N \cdot m \cdot s / radius )</td>
</tr>
<tr>
<td>( R_{Ja} )</td>
<td>Internal resistance of altitude motor</td>
<td>( \Omega )</td>
</tr>
<tr>
<td>( L_{Ja} )</td>
<td>Internal inductance of altitude motor</td>
<td>( H )</td>
</tr>
<tr>
<td>( K_{Ja} )</td>
<td>Motor constant of altitude motor system</td>
<td>( N \cdot m / Amps )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>States</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_{Ja} )</td>
<td>Angular position of altitude point</td>
<td>Radius</td>
</tr>
<tr>
<td>( \theta_{Ja} )</td>
<td>Altitude angular position of OTA</td>
<td>Radius</td>
</tr>
<tr>
<td>( \theta_{Ja} )</td>
<td>Altitude angular position of mirror mount</td>
<td>Radius</td>
</tr>
<tr>
<td>( \theta_{Ja} )</td>
<td>Altitude angular position of fork around z axis</td>
<td>Radius</td>
</tr>
<tr>
<td>( \Omega_{Ja} )</td>
<td>Altitude angular velocity of altitude point</td>
<td>( Radius / s )</td>
</tr>
<tr>
<td>( \Omega_{Ja} )</td>
<td>Altitude angular velocity of OTA</td>
<td>( Radius / s )</td>
</tr>
<tr>
<td>( \Omega_{Ja} )</td>
<td>Altitude angular velocity of mirror mount</td>
<td>( Radius / s )</td>
</tr>
<tr>
<td>( \Omega_{Ja} )</td>
<td>Altitude angular velocity of fork</td>
<td>( Radius / s )</td>
</tr>
<tr>
<td>( i_{Ja} )</td>
<td>Current through altitude control motor</td>
<td>Amps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{Ja} )</td>
<td>Input voltage for altitude control motor</td>
<td>( V )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{Ja} )</td>
<td>Wind disturbance act on OTA for altitude model</td>
<td>( N \cdot m )</td>
</tr>
<tr>
<td>( T_{Ja} )</td>
<td>Wind disturbance act on mirror mount for altitude model</td>
<td>( N \cdot m )</td>
</tr>
<tr>
<td>( T_{Ja} )</td>
<td>Wind disturbance act on fork</td>
<td>( N \cdot m )</td>
</tr>
<tr>
<td>( T_{Ja} )</td>
<td>Torque from gravitational force on OTA</td>
<td>( N \cdot m )</td>
</tr>
<tr>
<td>( T_{Ja} )</td>
<td>Torque from gravitational force on mirror mount</td>
<td>( N \cdot m )</td>
</tr>
</tbody>
</table>
Table 6: Modeling parameters for Altitude Model (Part2)

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{J,a}$</td>
<td>Angular position of altitude as output</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$\Omega_{J,a}$</td>
<td>Altitude angular velocity</td>
<td>$\text{Radius}$</td>
</tr>
<tr>
<td>$Y_{i,a}$</td>
<td>Guide star feedback sensor for altitude</td>
<td>$\text{Radius}$</td>
</tr>
</tbody>
</table>

2.4.2: Altitude Model State Equations

Figure 11: Altitude Telescope Model

Graphical technique is used to represent dynamics system model [5]. Linear graph represent the topological relationships of lumped-element interconnections within system [5]. Linear graph of altitude model is shown
in Figure 12. This graph is used to represent both the electrical and mechanical system. The loop on the left is the model of the DC motor. This part is identical to the azimuth system. The bigger loop on the right is the model of the telescope mechanical system. It is different from the azimuth due to its different structure. The component connecting the two systems is electric mechanical component motor. The motor serves as a transformer in between two loops to transfer current (shown as 1 on Figure 12) to torque (shown as 2 on Figure 12). From the linear graph, the system has 8 states (Shown in Equation 3). These 8 states are used to generate 8 state equations to represent dynamic motion in the azimuth system.

Equation 19: States of altitude system

\[ X = \begin{bmatrix} \Omega_{f_a}, \Omega_{f_o}, \Omega_{m}, \Omega_{m_o}, T_{K_f}, T_{f_o}, T_{m}, T_{m_o}, I_{L_a} \end{bmatrix} \]

Eight state variables are used to generate 8 state equations to represent dynamic motion in the altitude system (Shown in Equation 20). The detail derivation is shown in Appendix B.
Figure 12: Altitude Telescope Model Tree Diagram
Equation 20: State equation of altitude system

\[
\frac{d\Omega_{j,a}}{dt} = \frac{1}{J_{j,a}} \left[ -\frac{1}{K_{a,a}} i_{L,a} - B_{o,a} (\Omega_{j,a} - \Omega_{j,a}') - T_{K_o,a} - B_{o,a} (\Omega_{j,a} - \Omega_{j,o,a}) - T_{K_m,a} - B_{m,a} (\Omega_{j,a} - \Omega_{j,m,a}) \right]
\]

\[
\frac{d\Omega_{j,f,a}}{dt} = \frac{1}{J_{f,a}} \left[ -\frac{1}{K_{a,a}} i_{L,a} + B_{o,a} (\Omega_{j,a} - \Omega_{j,f,a}) - T_{K_f,a} - B_{f,a} \Omega_{j,f} + T_{s} \right]
\]

\[
\frac{d\Omega_{j,o,a}}{dt} = \frac{1}{J_{o,a}} \left[ T_{K_o,a} + B_{o,a} (\Omega_{j,a} - \Omega_{j,o,a}) + T_{s_{1,a}} + T_{s_{4,a}} \right]
\]

\[
\frac{d\Omega_{j,m,a}}{dt} = \frac{1}{J_{m,a}} \left[ T_{K_m,a} + B_{m,a} (\Omega_{j,a} - \Omega_{j,m,a}) + T_{s_{2,a}} + T_{s_{5,a}} \right]
\]

\[
\frac{d\omega_{l,a}}{dt} = \frac{1}{L_{o,a}} (V_s - R_i_L - \frac{1}{K_a} (\Omega_{j,a} - \Omega_{j,f,a}))
\]

\[
\frac{dT_{K_f,a}}{dt} = \frac{1}{J_{f,a}} \Omega_{j,f,a}
\]

\[
\frac{dT_{K_o,a}}{dt} = \frac{1}{J_{o,a}} (\Omega_{j,a} - \Omega_{j,o,a})
\]

\[
\frac{dT_{K_m,a}}{dt} = \frac{1}{J_{m,a}} (\Omega_{j,a} - \Omega_{j,m,a})
\]

Four states $T_{K_f,a}, T_{K_o,a}, T_{K_m,a}$ shown in Equation 19 are hard to be measured in the altitude system with accessible sensors. However, it is easier to transform them into a different format. Torque between two components can be rewritten as angular displacement multiplied by the rotational spring constant (Shown in Equation 21). $\theta_f$ defines angular displacement of fork; therefore, torque in fork is only a function of $\theta_f$ and $K_f$. 
Equation 21: Torque transformation of altitude model

\[
T_{kf\_a} = K_{f\_a} \theta_{f\_a} \\
T_{ko\_a} = K_{O\_a} \left( \theta_{j\_a} - \theta_{O\_a} \right) \\
T_{km\_a} = K_{m\_a} \left( \theta_{j\_a} - \theta_{m\_a} \right)
\]

Altitude model states have now became angular positions and angular velocities of system components and current in the electric system. Altitude models states is shown in Equation 22. With the new state variables, state equation is transformed also. New state equation is shown in Equation 23.

Equation 22: Altitude Model State

\[
X = \begin{bmatrix}
\theta_{j\_a} \\
\theta_{f\_a} \\
\theta_{jO\_a} \\
\theta_{jm\_a} \\
\Omega_{j\_a} \\
\Omega_{f\_a} \\
\Omega_{jO\_a} \\
\Omega_{jm\_a} \\
i_{L\_a}
\end{bmatrix}
\]
Equation 23: Transform state equations for altitude model

\[
\begin{align*}
d\theta_{j,a} &= \Omega_{j,a} \\
d\theta_{f,a} &= \Omega_{f,a} \\
d\theta_{o,a} &= \Omega_{o,a} \\
d\theta_{m,a} &= \Omega_{m,a} \\
d\Omega_{j,a} &= \frac{1}{J_{j,a}} \left[ \frac{1}{K_{a,a}} \left( i_{L,a} - B_{a} (\Omega_{j,a} - \Omega_{f,a}) - K_{a,a} (\theta_{j,a} - \theta_{f,a}) - B_{a} (\Omega_{j,a} - \Omega_{f,a}) \right) \
&\quad - K_{m,a} (\theta_{j,a} - \theta_{m,a}) - B_{m,a} (\Omega_{j,a} - \Omega_{m,a}) \right] \\
d\Omega_{f,a} &= \frac{1}{J_{f,a}} \left[ - \frac{1}{K_{a,a}} \left( i_{L,a} + B_{a} (\Omega_{j,a} - \Omega_{f,a}) - K_{f,a} \theta_{f,a} - B_{f,a} \Omega_{f} + T_{s3} \right) \right] \\
d\Omega_{o,a} &= \frac{1}{J_{o,a}} \left[ K_{a,a} (\theta_{o,a} - \theta_{f,o,a}) + B_{a} (\Omega_{j,a} - \Omega_{o,a}) + T_{s1,a} + T_{s4,a} \right] \\
d\Omega_{m,a} &= \frac{1}{J_{m,a}} \left[ K_{m,a} (\theta_{j,a} - \theta_{m,a}) + B_{m,a} (\Omega_{j,a} - \Omega_{m,a}) + T_{s2,a} + T_{s5,a} \right] \\
di_{L,a} &= \frac{1}{L_{a}} \left( V_{s} - Ri_{L,a} - \frac{1}{Ka} (\Omega_{j,a} - \Omega_{f,a}) \right)
\end{align*}
\]

2.4.3: State-space Model of Altitude System

8 states variables are the minimum set of variables which can be used to describe the altitude system. System outputs are system variables that are accessible and can be measured by the sensors (shown in Equation 26: Altitude Model Output). There are 3 system outputs we can obtain from sensors. They can generate 3 output equations. 8 state variables are substituted with 9 state variables. These 9 state equations and output equations are rewritten in state-space format (shown in Equation 24).
Equation 24: State Space Model for Altitude

\[
\begin{align*}
\dot{X}_a &= A_{a9}X_a + B_{a9}U_{a1} + E_{a9}d_{a5} \\
\dot{Y}_a &= C_{a3X}X_a + D_{a3}U_{a1}
\end{align*}
\]

The system input of altitude model is voltage sent to the electric system (Shown in Equation 24).

Equation 25: Altitude Model Input

\[
U = \begin{bmatrix} V_{\alpha} \\ Y_{\alpha} \end{bmatrix}
\]

There are 3 system outputs. Angular position and angular velocity of motor could be obtained from altitude encoder. \( Y_{\alpha} \) is the output from star image sensor. It is the displacement of reference star.

Equation 26: Altitude Model Output

\[
Y = \begin{bmatrix} \theta_{\alpha} \\ \Omega_{\alpha} \\ Y_{\alpha} \end{bmatrix}
\]

\( Y_{\alpha} \) is a function of mirror mount angular position and OTA angular position of the azimuth system (shown in Figure 10). Its relation is listed in Equation 27. Using small angle approximation, Equation 27 can be linearized into a simpler format (shown in Equation 29). The equation is different with the azimuth model. Fork vibration affects altitude pointing.
Equation 27: Star Image Displacement Calculation for altitude model
\[ Y_{t,a} = L_{OTA} \times \sin\{2[(\theta_{Jo,a} - \theta_{Jo,a} + \theta_{Jf,a} + \theta_{Jf,a})] - L_{OTA} \times \sin[(\theta_{Jo,a} - \theta_{Jo,a} + \theta_{Jf,a} + \theta_{Jf,a})] \]

Equation 28: Reduced Star Image Displacement Equation of altitude model
\[ Y_{t,a} = -L_{OTA} \theta_{Jo,a} + L_{OTA} \theta_{Jf,a} - L_{OTA} \theta_{Jo,a} + 2L_{OTA} \theta_{Jm,a} \]

There are five disturbance sources in the altitude model (Shown in Equation 29). The wind torque applied on OTA, the wind torque applied on mirror mount and the wind torque applied on the fork. Another two disturbances are gravitational disturbance on OTA and mirror mount.

Equation 29: Altitude Model Disturbance Source
\[
\begin{bmatrix}
T_{s1,a} \\
T_{s2,a} \\
T_{s3,a} \\
T_{s4,a} \\
T_{s5,a}
\end{bmatrix}
\]

All system matrices can be determined by transformer 9 state equations and 3 output equations. System matrices are shown below. A_a matrix shown in Equation 14 is the state matrix of the altitude system.
**Equation 30:** System \((A_a)\) matrix for altitude model

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-\left(\frac{K_{O_a}+K_{m_a}}{J_{m_a}}\right) & 0 & \frac{K_{O_m}}{J_{m_a}} & \frac{K_{m_a}}{J_{m_a}} & -\left(B_{a}+B_{O_a}+B_{m_a}\right) & \frac{B_{a}}{J_{m_a}} & \frac{B_{O_m}}{J_{m_a}} & \frac{B_{m_m}}{J_{m_m}} & 1
\end{bmatrix}
\]

The \(B_a\) matrix shown in Equation 31 is the input matrix of the altitude system.

**Equation 31:** Input \((B_a)\) matrix for altitude model

\[
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[
L_{a}^{9X1}
\]

The \(C_a\) matrix shown in Equation 32 is the output matrix of the altitude system. Output variables \(\theta_{j_a}\) and \(\Omega_{j_a}\) is a direction output from state variables.

\(Y_{j_a}\) is calculated using Equation 28.
Equation 32: Output (C_a) matrix for altitude model

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}_{3 \times 9}
\]

D_a matrix shown in Equation 33 is the direct transmittance matrix for the altitude model. In the altitude system, input of the system doesn't affect outputs; therefore the transmittance matrix is a zero matrix.

Equation 33: Direct transmittance (D_a) matrix for altitude model

\[
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}_{3 \times 1}
\]

E_a matrix shown in Equation 18 is the disturbance matrix for the altitude system.
Equation 34: Disturbance ($E_a$) matrix for altitude model

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{J_f} \\
0 & \frac{1}{J_o} & 0 & 0 & \frac{1}{J_o} \\
0 & \frac{1}{J_m} & 0 & 0 & \frac{1}{J_m} \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}_{9 \times 5}
\]

2.5: Matlab Simulation

2.5.1: Simulink Model

In order to simulate the telescope motion, the state-space formulas are imported into Simulink. A state space subsystem is built for handling the state space formula. The state-space subsystem is shown in Figure 13. This subsystem integrates the change rate of last step and adds the saved current state position. This forms the current state position. The current position and input are used to calculate the change rate of the current step and then import into the next step.
Gravity disturbances and wind disturbance is calculated with environmental wind speed and states of telescope. Because of the models are nonlinear, they are calculated externally in a different subsystem and imported into the state-space loop as disturbance inputs. Gravity model is shown in Figure 14. Wind model for altitude is shown in Figure 15.
The full flexible telescope simulator is shown in Figure 16. It consists of a two state-space handler for the azimuth system and the altitude system and a disturbance model. Disturbance model includes gravity model and two wind models for the azimuth model and the altitude model. A complete set of
telescope Simulink model can be found in Appendix C.

Disturbance model takes wind direction, wind speed and telescope pose from state variables as input. It generates seven disturbance outputs and sent to the azimuth and the altitude model.

![Simulink model of telescope](image)

*Figure 16: Full telescope in Simulink*

Two PID controllers are added into the system to perform position feedback control using angular position outputs from the azimuth model and the altitude model. PID control is shown in Figure 17.
2.5.2: Simulation Output

The result from the simulation is saved in Matlab. There are three Matlab plotting functions for handling a saved result. “Plot_Function_Azimuth.m” is used to plot azimuth motion. The result is shown in Figure 18. The top plot on the left shows the telescope point at the rotational axis. The mid plot on the left shows feedback from star sensor, which represents the combination telescope pointing. The bottom plot on the left shows the control input sent to telescope. Three plots on the right shows vibration of fork, OTA and mirror mount of azimuth system. The vibration is defined as difference between component angular position and main telescope pointing direction.

![PID controller diagram](image.png)
"Plot_Function_Altitude.m" is used to plot altitude motion. The result is shown in Figure 19. It plots same information as azimuth motion.
Figure 19: Result of altitude Simulation

“Plot_Function_2Axis” shows displacement on two main axes and feedback on star image sensor. This information is actually observable to the control system. It also plots the change path of combined telescope pointing direction and star image feedback on XY coordination during telescope motion. The result is shown in Figure 20. The detail plotting function can be found in Appendix E.
The telescope pointing direction from the image feedback sensor shows a higher frequency motion than the pointing direction obtained from encoder. This is because the image feedback sensor also reflects the vibration in the OTA and the mirror mount.
Chapter 3: Azimuth Experimental Telescope

In order to verify the simulation, a one degree of freedom experimental telescope is built to generate a set of experimental data. The simulation parameters are adjusted so that the simulation result can reassemble this experimental data.

3.1: Experiment Set Up

3.1.1: Telescope Structure

The experimental test stand reproduces the azimuth model. The telescope experimental test stand is shown in Figure 21. The frame of experimental test stand is considered as part of the ground and not considered in the model. A Pittman 14203 motor with an E60 encoder is mounted at the bottom plate between two beams of the frame. The telescope body, composited of long steel plates, is mounted between the top and bottom plates. In order to reduce fiction, two shaft adapters are manufactured using the CNC machine to mount the telescope body. The main reflector is taken down from an ORION telescope StarBlast 4.5. Its diameter is 113 mm and has a focal length of 450 mm.
Analog output from the controller is amplified using the MOTOMATIC Station used for Cal Poly, San Luis Obispo mechanical engineering ME 422 control class (Shown in Figure 22). The power amplifier saturated at 3 amps output. The green LED on the MOTOMATIC station turns red when power the amplifier is saturated. The power amplified has a gain of 5.25 V/V.

Derivative control is not used because the resolution of the E60 encoder attached to the Pittman motor is low. E60 encoder has 1024 pulses per revolution. Its accuracy is only 0.35 degree. Derivation of feedback error becomes infinite when encoder increment. This produces a high frequency noise in output signal when the telescope runs at high speed and produces a nonlinear system behavior during low speed. This behavior cause instability of telescope test stand; therefore, derivative gain is set to 0 for position controller of
The experimental is controlled using a position control with PI feedback controller. The controller is constructed using Simulink. Control signals are generated by SC-2075 DAQ Interface (Data Acquisition Interface). SC-2075 DAQ interface is shown in Figure 23). This DAQ interface connects to an internal National Instrument E-Series DAQ card. The controller uses a bidirectional counter embedded in the E-Series DAQ card to determine the angular displacement of motor.

Current position controller used for the system is a single input single output controller. It takes either motor encoder reading or the star image sensors reading as input, and sends an analog signal as output. The output is connects to
power amplifier input on the MOTOMATIC station.

![SC-2075 DAQ Interface](image)

**Figure 23: SC-2075 DAQ Interface**

### 3.1.3: Guide Star Feedback Simulation

The star source is simulated with a 100W light bulb. The feedback sensor is most sensitive to infrared and red light; therefore an incandescent lamp is used for this experiment. The simulated source is shown in Figure 24.
Light from the simulated source shines on the mirror and is reflected to the sensor placed close to focal point of the mirror. The star image sensor is simulated with two LED set next to each other. The distance between the two LED is adjusted so it is the same as the diameter of source light. An instrument amplifier is used to take the differential between voltage outputs of LED sensors. When the light shines on LED, the LED generates an electric current according to the intensity of the light shines on it. A light sensor used on the test stand is shown in Figure 25. The star feedback is shining on the right LED on this figure.
Figure 26: Star image feedback simulation sensor situation

Figure 26 shows the layout of the LED sensor. At part (A), the light source shines at the center of two sensors. Two LED send the same signal; therefore, the differential reading of two LEDs is 0. At part (B), the light source shines at the right LED. The right LED has a high reading and the left LED has low reading; therefore, the differential signal is positive. At part (C), the signal is negative.

The output from the star image sensor is shown in Figure 27. The feedback contains many high frequency noises in it. A low pass filter is applied in order to make the signal stable. In order to use this feedback in a leaner system, the center part (from 6.62 to 6.97s) of the signal is linearized. The response of the star image sensor is treated as a linear response with the relationship of the sensor output and the telescope angular position between 6.62 and 6.97 seconds. This time region is the stable region of the feedback system. If
telescope intended to use the simulated star image, the system goes unstable. For the real telescope control system, a more complex feedback system is required to judge the position of the reference star.

Figure 27: Light response from Simulation

3.2: Experiments Result

In order to obtain the system behavior of the experimental test stand, four tests are conducted. The testing setup is shown in Figure 28.
3.2.1: Light Feedback

Light feedback test is performed to find the relationship between the star image feedback and the angular position of the telescope. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Input”. The instruction about running real time window target is shown in program interface.

2. Move telescope to find the lowest signal output.

3. Move telescope to increase signal reading and record angular displacement.

4. Repeat procedure (3) until reach highest output signal reading.
The result of the light feedback test is shown in Figure 29. This result is used to calibrate the controller gain.

The result shows that the relation between signal output and the angular position of the telescope is close to linear; however, there is a small voltage offset. When the sensor points away from the simulated source, which the differential reading should be 0, the reading is about 20 mV.

Figure 29: Star Image feedback reading
3.2.2: PI Control

The PI control test is performed to understand telescope response using the encoder as position feedback. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Telescope_Control”. The instruction about running the real time window target is shown in program interface.

2. Adjust telescope body to center.

3. Run the program “Telescope_Control”

4. Adjust PI gain and run program again.

The PI gains that yield the best result: Kp is 7 and Ki is 0.6. The proportional gain is picked so the system has a settling time shorter than 10 sec. The integral is picked to minimize static state error. The result is shown in Figure 30. The control signal in the figure is signal sent to the power amplifier. It is required to multiply the power amplifier gain to obtain actual signal sent to motor. The response didn’t actually go to the desired position. The small error toward the end is due to high static friction in the system. It is shown that the integral gain is increasing trying to compensate for that error.
3.2.3: Tap Test

The tap test is performed to find the frequency of the experimental test stand. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Input”. The instruction about running real time window target is shown in the program interface.

2. Adjust the telescope body to center.

3. Clamp the center shaft of the telescope to limit its motion.
4. Gently tap the end of the OTA 3 times.

5. Record the response of the telescope sensor.

Due to the low resolution of the motor encoder, this response isn’t showing in the encoder feedback. The feedback is obtained from the star image sensor. The result is shown in Figure 31. A low pass filter is applied to the signal. The result is shown in the lower part of same figure. The three peaks represent the three tap input to the system during the test. The response shows that the vibration is slowly damped out.
In order to understand the frequency response of telescope, A FFT (fast Fourier transform) is used to analyze signal feedback. FFT function from Matlab is used for this analysis. The result is shown in Figure 32. The result shows a much higher response density on the lower frequency region. Three major response frequencies occur at 5Hz, 15Hz and 25 Hz. Two other responses show
at 60Hz and 175Hz. Further experiment is required to identify the source of each frequency.

![OTA Response](image)

**Figure 32: Frequency response of telescope to tap test**

### 3.2.4: Feedback Control Using Light Feedback

Feedback control using light feedback is performed to understand telescope behavior when the star image sensor is used as feedback.
signal. PI control is used in this test. The experimental procedure is shown below.

1. Connect the experimental test stand to the computer and run the program “Telescope_Control_Light”. The instruction about running the real time window target is shown in the program interface.

2. Adjust the telescope body to center. Make sure source light shines at the center of two sensors.

3. Create an offset by moving telescope pointing slightly. Pointing direction shouldn’t exceed linear region of feedback signals.

4. Run the program “Telescope_Control_light”

5. Adjust PI gain and run program again.

The result of light feedback control is shown in Figure 33. The output signal from star image sensor is calibrated to show angular displacement in degree. Three different vibration frequencies are visible from the signal feedback. A FFT (fast Fourier transform) is used to analyze signal feedback. The result is shown in the third plot of Figure 33.
The result shows highest density toward the low frequency region. This peak comes from the oscillation of telescope motion with used control gains. The next peak shows up at 9Hz. This vibration frequency also shows up in the tap test. The source of this vibration is OTA. During the test, it is visible that OTA has a vibration motion with frequency higher than main telescope motion. Another peak shows up at 120Hz. It is not clear if it is the response from the mirror mount or the noise from AC power of the system. It is possible that the source is the
mirror mount vibration. This peak does not show in the tap test because a tap on OTA does not excite the mirror mount much. The input from control motor excites both OTA and mirror mount equally. Other frequencies where peaks occur in tap are not shown in this test because the vibration from OTA and the mirror mount is dominating. The calculation result of the natural frequency of the OTA is 7.00 Hz. The calculation result of the natural frequency of the mirror mount is 10.70 Hz. The calculation is shown in Appendix F. This result shows that the frequency is really close. Even though the mirror mount is a lot stiffer than OTA. The vibration frequency is close because the main reflector mounted on the mirror mount is a lot heavier than the instrument mounted on the OTA.

The second plot shows telescope pointing direction in degree. It shows that star image sensor works. The displacement of telescope is about 1 degree. The E60 encoder only has 3 step reading in this amount of displacement. It is hard to control small displacement using encoder as feedback source. Star image is useful for fine adjustment of telescope pointing.
Chapter 4: Simulation Result Analysis

In order to test the telescope model's ability to reassemble a real system, a similar test performed on the experimental test stand is simulated with the state space model of telescope. Full telescope model is reduced to the azimuth model, and disturbances are set to 0.

4.1: Turned Simulation model

4.1.1: PI Control Tuning

The telescope azimuth model is used to simulate PI control response of the experimental telescope. A set of general parameters is used to run the simulation first. The model is turned based on response shown in Figure 30. The PI control simulation result of turned model is shown in Figure 34. The system parameter is shown in Appendix G.
4.1.2: Frequency Response of Simulation Model

A pulse input is given to the model in order to simulate tap test. The result is shown in Figure 35. The first plot on the left is the output on the star image sensor. It shows similar patent as Figure 33. A FFT (fast Fourier transform) is used to analyze simulated signal feedback. Time domain responses of telescope components are shown on the left. 4 plots on the right show the result of FFT analysis of each component. The result shows highest density toward the low frequency region. This peak comes from the oscillation of telescope motion with used control gains. The simulated response oscillates less; therefore, the density is smaller comparing with test result. Next peak shows up at 30Hz. The source of this is OTA vibration. This peak is also shown at OTA frequency
response plot. Next peak shows at 68Hz. The source is a mirror mount vibration. This peak is also shown at mirror mount frequency response plot. Vibrations from OTA and mirror mount project its own vibration to each other. There is a peak with small density shown on the mirror mount frequency response plot at the location where OTA vibration occurs.

![Figure 35: Frequency response of simulation model]

The vibration frequency of OTA and mirror mount and OTA is show in Table 7. The natural frequencies of components match frequencies show on the frequency response plot. Natural frequency of component is adjustable by changing component mass and spring constant. The performance of simulation can be turned to fit the real system.
Table 7: Natural frequency of components

<table>
<thead>
<tr>
<th>Spring Constant [radius/Nm]</th>
<th>Inertia [Kgm^2]</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mirror Mount</td>
<td>1806</td>
<td>0.0103</td>
</tr>
<tr>
<td>OTA</td>
<td>774</td>
<td>0.0298</td>
</tr>
</tbody>
</table>

4.2: Controllability and Observability Analysis

With system parameters that generates a simulated result that similar to real system, controllability and observability of system is analyzed. A linear system is controllable if it is possible to find a control input which can transfer a state to the origin at some finite time. A linear system is observable if its states can be determined from the output function. Controllability and observability is analyzed using criterion 2[7]. The result is shown in Table 8.

Table 8: Controllability and Observability Analysis

<table>
<thead>
<tr>
<th></th>
<th>Controllability</th>
<th>Observability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth System</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Altitude System</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

It shows that the system is not fully controllable not fully observable. The states that are controllable and observable in the altitude system are the angular position and the angular velocity of motor: \( \theta_{J_{a}}, \Omega_{J_{a}} \). The states that are controllable in the azimuth system are the angular position and the angular velocity of the azimuth motor and the angular position of fork: \( \theta_{j}, \theta_{jam}, \Omega_{j} \). The states that are observable in the azimuth system are the angular position and the angular velocity of the azimuth motor: \( \theta_{j}, \Omega_{j} \). The angular position of the fork is
not observable. The star image feedback gives a combination of all vibration, but it does not show the actual position and velocity of each components.
Chapter 5: Conclusion

5.1: Light Weight Flexible Telescope Model

This project considers a light weight flexible structure telescope. By using a directive drive, it is possible to use active feedback control to reduce vibration of the telescope. This thesis fits into a larger project with a goal of controlling a flexible telescope in order to help a cheaper telescope achieve higher accuracy. In order understand the internal motion of the telescope, the goal of this thesis is to build a state space dynamic model of a two degree of freedom telescope using a known star location as feedback. The model consists of structure model with two inputs and two major environments. Two inputs are control inputs for the azimuth motor and the altitude motor. Source of disturbances are gravity and wind. The result of model is shown in Figure 16.

5.2: Experimental Test Stand

A one degree freedom experimental test stand is built to generate a series of telescope motion data in order to exam that if the telescope simulation model can represent a real telescope system. For recording the telescope motion, a wheel encoder mounted on the azimuth motor and a LED light density sensor mounted on the end of telescope body are used as feedbacks. A light feedback test is performed to find the relationship between the image sensor output and the angular position of the telescope. A tap test is performed to find the response
frequency of the experimental test stand. Two feedback position controls using the wheel encoder and the light feedback sensor are performed to observe the telescope behavior with different feedback system. The result shows that the simulation model can reproduce many system patterns shown in the experimental data; however, it is not feasible to obtain all of telescope's physical parameters from the system feedbacks.

5.3: Control Analysis

With system parameters that generates a simulated result that similar to real system, controllability and observability of system is analyzed. It shows that the system is not fully controllable or fully observable. The controllability and observability is shown in Table 8. The states that are controllable and observable in the altitude system are the angular position and the angular velocity of motor: $\theta_{j-a}, \dot{\theta}_{j-a}$. The states that are controllable in the azimuth system are the angular position and the angular velocity of the azimuth motor and the angular position of fork: $\theta_j, \dot{\theta}_{j-a}, \Omega_j$. The states that are observable in the azimuth system are the angular position and the angular velocity of the azimuth motor: $\theta_j, \Omega_j$. The angular position of the fork is not observable. The star image feedback gives a combination of all vibration, but it does not show the actual position and velocity of each components.

However, it does not mean it isn’t possible to actively reduce vibration in the telescope. The current dynamic model contains three vibration
elements. In between them, the OTA causes the greatest vibration; therefore, the controller should be designed to compensate the OTA vibration. The observability is a bigger issue for controlling the system. The controller can assume the feedback from the star image sensor is caused by the OTA vibration and use it to compensate the OTA vibration. However, the experiment shows that the mirror mount could have a natural frequency close to the OTA natural frequency. It is possible that the input for reducing the OTA vibration will amplify the mirror mount vibration. To avoid this situation, more sensors are required in order to observe the system.

5.4: Current Project State and Future Work

In this project, a state space model of a light weight flexible telescope is built. The model will be used for the telescope simulation and the control design in the future. The state space formula also enables it to perform multiple inputs and multiple outputs control. However, due to the limitation of experimental test stand, more work should be done to understand the behavior of a light weight flexible telescope. The azimuth motor needs a wheel encoder with better resolution. The encoder should at least have 0.1 degree resolution for a better test result. A light sensor array should be built to give a bigger range of star image displacement feedback. These improvements will help understand the telescope response and build a better controller.
Reference


Appendix A: Derivation of Azimuth Model
\[ \frac{d\Omega_J}{dt} = \frac{1}{J} T_J \]

\[ T_J = -T_a - T_b - T_{kt} - T_{bt} \]

\[ \frac{d\Omega_{Jam}}{dt} = \frac{1}{J_{am}} T_{Jam} \]

\[ T_{Jam} = T_{kf} + T_{bf} - T_{ao} - T_{am} - T_{an} - T_{bm} \]

\[ \frac{d\Omega_J}{dt} = \frac{1}{J_o} T_J \]

\[ T_J = T_{ko} + T_{bo} + T_{so} \]

\[ \frac{d\Omega_{Jm}}{dt} = \frac{1}{J_m} T_{Jm} \]

\[ T_{Jm} = T_{maj} + T_{am} + T_{bm} \]

\[ \frac{dL}{dt} = \frac{1}{L} V_L \]

\[ V_L = V_s - V_R - V_I \]

\[ \frac{dT_{kf}}{dt} = K_f \Omega_{kf} \]

\[ \Omega_{kf} = \Omega_{J} - \Omega_{Jam} \]

\[ \frac{dT_{ko}}{dt} = K_o \Omega_{ko} \]

\[ \Omega_{ko} = \Omega_{Jam} - \Omega_{Jm} \]

\[ \frac{dT_{km}}{dt} = K_m \Omega_{km} \]

\[ \Omega_{km} = \Omega_{Jm} - \Omega_{Jm} \]

\[ T_b = B \Omega_J = B \Omega_J \]

\[ T_{bf} = B_f \Omega_{bf} = B_f (\Omega_J - \Omega_{Jam}) \]

\[ T_{bo} = B_o \Omega_{bo} = B_o (\Omega_{Jam} - \Omega_{Jm}) \]

\[ T_{bm} = B_m \Omega_{bm} = B_m (\Omega_{Jm} - \Omega_{Jm}) \]

\[ V_R = R \dot{\lambda} \]

\[ V_I = \frac{1}{K_m} \Omega_J = \frac{1}{K_m} \Omega_J \]

\[ T_{ao} = -\frac{1}{K_a} \dot{\lambda} \]

Replace state with

\[ \theta_J, \theta_{Jam}, \theta_{Jo}, \theta_{Jm} \]

\[ \Omega_{J}, \Omega_{Jam}, \Omega_{Jo}, \Omega_{Jm} \]

\[ \dot{\lambda} \]

Replace \( T_{kf}, T_{ko}, T_{km} \)

\[ T_{kf} = K_f (\theta_J - \theta_{Jam}) \]

\[ T_{km} = K_m (\theta_{Jam} - \theta_{Jm}) \]

\[ T_{ko} = K_o (\theta_{Jam} - \theta_{Jm}) \]
STATE VARIABLES: $\Theta_j, \Theta_{j,\alpha}, \Theta_i, \Theta_{i,\alpha}, S_j, \Omega_{j,\alpha}, \Omega_i, \Omega_{i,\alpha}$

STATE

$X = \begin{bmatrix}
\Theta_j \\
\Theta_{j,\alpha} \\
\Theta_i \\
\Theta_{i,\alpha} \\
S_j \\
\Omega_j \\
\Omega_{j,\alpha} \\
\Omega_i \\
\Omega_{i,\alpha} \\
\dot{x}_L
\end{bmatrix}_{9 \times 1}$

INPUT

$U = \begin{bmatrix} U_b \end{bmatrix}_{1 \times 1}$

DISTURBANCE

$d = \begin{bmatrix} T_{e1} \\ T_{e2} \end{bmatrix}_{2 \times 1}$

OUTPUT

$Y = \begin{bmatrix} \Theta_j \\ \Omega_j \\ \dot{x}_L \end{bmatrix}_{3 \times 1}$

$\dot{X}_{9 \times 1} = A_{9 \times 9} X_{9 \times 1} + B_{9 \times 1} U_{1 \times 1} + E_{9 \times 2} d_{2 \times 1}$

$Y_{3 \times 1} = C_{3 \times 9} X_{9 \times 1} + D_{3 \times 1} U_{1 \times 1}$
DISTURBANCE

\[ d = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix} \]

\[ T_{s1} \text{ and } T_{s2} \text{ is the torque caused by wind that act on telescope body} \]

in azimuth model, it affects \( \Omega^2 \), \( \Omega \Omega^m \), affection on \( \Omega^2 \) is ignored, since rotation of fork due to wind is very small

Wind Model

Wind Pressure = \( \frac{1}{2} \times (\text{density of air}) \times (\text{Wind Speed})^2 \times (\text{Shape Factor}) \)

\[ V_w \]

density of air \( \approx 1.25 \text{ kg/m}^3 \)
shape factor \( = 1 \)

\[ \text{port of wind speed} = \frac{V_w}{3} \]

\[ T_{s1} = \frac{1}{2} \times 1.25 \times \frac{k_g}{m^2} \times \left[ V_w \sin(\theta_{5 m} - \theta_w) \right] \frac{m^2}{3^2} \times A_{out} \text{ m}^2 \times \frac{1}{2} \times L_{\Omega^m} \text{ m} \]

\[ T_{s2} = \frac{1}{2} \times 1.25 \times \int V_w \sin (\theta_{5 m} - \theta_w) d \theta_{5 m} \times A_{\text{mirror}} \times \frac{1}{3} \times L_{\text{mirror}} \text{ mm}\]

since \( T_1 \) and \( T_2 \) would make model non linear therefore, they would be calculated externally and input to model as \( T_{s1} \) and \( T_{s2} \)
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Telescope Structure
Azimuth Model

\[ \begin{bmatrix}
R \\
L
\end{bmatrix} \]

9x9
C matrix

$\theta_i = \theta_j$

$\Omega_j = \Omega_j$

$x_{azi} = \text{distance of guide star to center of close loop feedback sensor}$

$x_{azi} = L \sin \left[ 2 (\theta_{5m} + \phi_{Jnn} - \theta_5) \right] + L \sin (\theta_{5m} + \phi_{Jnn} - \theta_5)$

in order to perform controlling, $x_{azi}$ is projected back to global coordination to present direction telescope is actually point

$x_{azi} = L \sin (2 (\theta_{5m} + \phi_{Jnn} - \theta_5)) - L \sin (\theta_{5m} + \phi_{Jnn} - \theta_5)$

Use small angle approximation

$x_{azi} = 3L \phi_{Jnn} + L \phi_{Joo} + 2L \theta_{5m}$

C matrix

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 3L & L & 2L & 0 & 0 & 0 & 0
\end{bmatrix}
$$
Appendix B: Disturbance Model

Telescope Structure

Wind

\[ \text{Wind Pressure} = \frac{1}{2} \times \text{density of air} \times \text{(wind speed)}^2 \times \text{(slope factor)} \]

\[ T_{w1-a} = \frac{1}{2} \times 1.25 \times \left[ V_w \cos (\theta_T - \theta_w) \sin \theta_{za} \right]^2 \times A_{\text{on-a}} \times \frac{1}{2} L_{\text{ota}} \]

These terms are from the azimuthal model.

\[ T_{w2-a} = \frac{1}{2} \times 1.25 \times \left[ V_w \cos (\theta_T - \theta_w) \sin \theta_{\text{min-a}} \right]^2 \times A_{\text{minor-a}} \times \frac{1}{2} L_{\text{mirror}} \]

\[ T_{w3-a} = \frac{1}{2} \times 1.25 \times \left[ V_w \cos (\theta_T - \theta_w) \right]^2 \times A_{\text{fan-a}} \times \frac{1}{2} L_{\text{fan}} \]

Gravity

Torque due to OTA

\[ T_{oa} = -m_o a \times g \times \cos (\theta_{ao-a}) \times \frac{1}{2} L_{\text{ota}} \]

\[ T_{oa} = -m_{\text{minor}} a \times g \times \cos (\theta_{\text{min-a}}) \times \frac{1}{2} L_{\text{minor}} \]

Since all the torque would make the model nonlinear, they would be calculated as external disturbance source.
\[
\begin{align*}
\mathbf{d} \mathbf{a} &= \begin{bmatrix}
    T_{s1} - a \\
    T_{s2} - a \\
    T_{s3} \\
    T_{s4} \\
    T_{s5} \\
\end{bmatrix} \quad 8 \times 1 \\

d \Omega_{fa} &= \left[ \begin{array}{cc}
    & T_{s3} \\
\end{array} \right] \frac{1}{J_f} \\

d \Omega_{Jo,a} &= \left[ \begin{array}{cc}
    & T_{s1} - a + T_{s4} \\
\end{array} \right] \frac{1}{J_o} \\

d \Omega_{Jm,a} &= \left[ \begin{array}{cc}
    & T_{s2} - a + T_{s5} \\
\end{array} \right] \frac{1}{J_m} \\
\end{align*}
\]

↑ TERM FOR STRUCTURE MATRIX
$E_a$ matrix

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\frac{1}{J_p} & 0 & 0 & 0 & 0 \\
\frac{1}{J_o} & 0 & 0 & 0 & 0 \\
\frac{1}{J_m} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{J_m} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

$9 \times 5$
Appendix C: Derivation of Altitude Model

The spring and damping between motor, OTA, mirror, and fork are rotational and make angular displacement.

\[ N_d = 2 \]
\[ N_d = 9 \]
\[ B_1 = 9.2 \times 7 \]
\[ B_L = 11 \]

System order: 8

Primary variables: \( V_0(t), V_a, i_1, V_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, \Omega_f, \Omega_s, \Omega_b, T_{in}, T_{out}, \Omega_{in}, \Omega_{out} \)

Secondary variables: \( i_0(t), i_2, V_L, i_1, \Omega_2, \Omega_3, \Omega_4, T_8, T_9, \Omega_{in}, \Omega_{out}, \Omega_{in}, T_{in}, T_{out}, \Omega_{in}, \Omega_{out} \)

State variables: \( i_0, \Omega_{in}, \Omega_{out}, \Omega_{in}, \Omega_{out}, i_1, T_{in}, T_{out}, T_{out}, T_{in} \)
### Telescope Structure
#### Altitude Model

\[ \frac{d\Omega_{J \ell}}{dt} = \frac{1}{J_0} T_{J \ell} \]
\[ T_{J \ell} = -T_2 - T_B - T_{k_0} - T_{\beta_0} - T_{\kappa m} - T_{\kappa m} \]

\[ \frac{d\Omega_{J f}}{dt} = \frac{1}{J_f} T_{J f} \]
\[ T_{J f} = T_2 + T_B - T_{k f} - T_{B f} \]

\[ \frac{d\Omega_{J_0}}{dt} = \frac{1}{J_0} T_{J_0} \]
\[ T_{J_0} = T_{k_0} + T_{\beta_0} + T_{z_1} \]

\[ \frac{d\Omega_{J m}}{dt} = \frac{1}{J_m} T_{J m} \]
\[ T_{J m} = T_{k m} + T_{\beta m} + T_{z 2} \]

\[ \dot{V}_L = \frac{1}{L} V_L \]
\[ V_L = V_0 - V_R - V_1 \]

\[ \frac{d\Omega_{k f}}{dt} = K_f \Omega_{k f} \]
\[ \Omega_{k f} = \Omega_{J f} \]

\[ \frac{d\Omega_{k_0}}{dt} = K_0 \Omega_{k_0} \]
\[ \Omega_{k_0} = \Omega_{J} - \Omega_{J_0} \]

\[ \frac{d\Omega_{k m}}{dt} = K_m \Omega_{k m} \]
\[ \Omega_{k m} = \Omega_{J} - \Omega_{J m} \]

\[ T_B = B \Omega_B = B (\Omega_{J} - \Omega_{J f}) \]
\[ T_{B f} = B_f \Omega_{B f} = B_f \Omega_{J f} \]
\[ T_{B 0} = B_0 \Omega_{B 0} = B_0 (\Omega_{J} - \Omega_{J 0}) \]
\[ T_{B m} = B_m \Omega_{B m} = B_m (\Omega_{J} - \Omega_{J m}) \]

\[ V_R = R \dot{V}_L \]
\[ V_1 = \frac{1}{k_0} \Omega_2 = \frac{1}{k_m} (\Omega_{J} - \Omega_{J f}) \]
\[ T_2 = \frac{1}{k_0} \dot{V}_L \]
Telescope Structure

Altitude Model

\[ \frac{t_e}{t_{\text{ref}}} = \frac{e_t}{\text{ref}} \]

\[ \frac{e_t}{t_{\text{ref}}} = \frac{k_0 (\text{g} - \text{g}_{\text{ref}})}{t_{\text{ref}}} \]

\[ \frac{k_0 (\text{g} - \text{g}_{\text{ref}})}{t_{\text{ref}}} = \frac{k_f (\text{g} - \text{g}_{\text{ref}})}{t_{\text{ref}}} \]

\[ \frac{k_f (\text{g} - \text{g}_{\text{ref}})}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]

\[ \frac{t_0}{t_{\text{ref}}} = \frac{t_0}{t_{\text{ref}}} \]
Telescope Structure
Altitude Model

\[ T_{\text{fps}} = k_F \Theta_f \]
\[ T_{\text{k0}} = k_0 (\Theta_J - \Theta_{\text{so}}) \]
\[ T_{\text{km}} = k_m (\Theta_J - \Theta_{\text{sm}}) \]

\[ \frac{d\Omega_J}{dt} = \frac{1}{J_J} \left[ -\frac{1}{k_a} \dot{i}_L - B (\Omega_J - \Omega_{\text{Jf}}) - k_o (\Theta_J - \Theta_{\text{so}}) - B_o (\Omega_J - \Omega_{\text{so}}) - k_m (\Theta_J - \Theta_{\text{sm}}) - B_m (\Omega_J - \Omega_{\text{sm}}) \right] \]

\[ \frac{d\Omega_{\text{Jf}}}{dt} = \frac{1}{J_f} \left[ -\frac{1}{k_a} \dot{i}_L + B (\Omega_J - \Omega_{\text{Jf}}) - k_F \Theta_f - B_F \omega_{\text{Jf}} \right] \]

\[ \frac{d\Omega_{\text{Jso}}}{dt} = \frac{1}{J_{\text{so}}} \left[ k_o (\Theta_J - \Theta_{\text{so}}) + B_o (\Omega_J - \Omega_{\text{Jso}}) + T_{\text{s1}}(t) \right] \]

\[ \frac{d\Omega_{\text{Jsm}}}{dt} = \frac{1}{J_{\text{sm}}} \left[ k_m (\Theta_J - \Theta_{\text{sm}}) + B_m (\Omega_J - \Omega_{\text{Jsm}}) + T_{\text{s2}}(t) \right] \]
STATE VARIABLES: \( \theta_j \), \( \theta_f \), \( \theta_o \), \( \theta_{im} \), \( \Omega_j \), \( \Omega_f \), \( \Omega_o \), \( \Omega_{im} \), \( \lambda_l \)

PUT EQUATION INTO STATE SPACE FORM

STATE \( \dot{X} = \begin{bmatrix} \dot{\theta}_j \\ \dot{\theta}_f \\ \dot{\theta}_o \\ \dot{\theta}_{im} \\ \dot{\Omega}_j \\ \dot{\Omega}_f \\ \dot{\Omega}_o \\ \dot{\Omega}_{im} \\ \dot{\lambda}_l \end{bmatrix} \) \( \text{INPUT} = \begin{bmatrix} u_0 \end{bmatrix} \) \( \text{DISTURBANCE} = \begin{bmatrix} T_{r1} \\ T_{r2} \end{bmatrix} \) \( \text{OUTPUT} = \begin{bmatrix} \theta_j \\ \Omega_j \\ \theta_t \end{bmatrix} \)

\[ \dot{X}_{9 \times 1} = A_{9 \times 9} X_{9 \times 1} + B_{9 \times 1} U_{1 \times 1} + E_{9 \times 2} D_{2 \times 1} \]

\[ \gamma_{3 \times 1} = C_{3 \times 9} X_{9 \times 1} + D_{3 \times 1} U_{1 \times 1} \]
DISTURBANCE

\[ d = \begin{bmatrix} T_{s1} \\ T_{s2} \end{bmatrix} \]

\( T_{s1} \) and \( T_{s2} \) is the torque caused by wind that act on telescope body.

In azimuth model, it affects \( \Omega_{im} \), \( \Omega_{im} \), affecting on \( \Delta T_{f} \) is ignored, since rotation of fork due to wind is very small.

Wind Model

Wind Pressure = \( \frac{1}{2} \times \text{density of air} \times (\text{Wind Speed})^2 \times V_{w} \times \text{Shape Factor} \)

Density of air \( \approx 1.25 \text{ kg/m}^2 \)

Shape factor = 1

Unit of wind speed = \( \frac{m}{s} \)

\( T_{s1} = \frac{1}{2} \times 1.25 \times \frac{m^2}{m^2} \times [V_{w} \sin(\theta_{10} - \theta_{20})] \frac{m^2}{s^2} \times A_{ota} \text{ m}^2 \times \frac{1}{2} L_{ota} \text{ m} \) \( \text{(15)} \)

\( T_{s2} = \frac{1}{2} \times 1.25 \times [V_{w} \sin (\theta_{im} - \theta_{i1})]^2 \times A_{mirror} \text{ m}^2 \times \frac{1}{2} L_{mirror} \text{ m} \) \( \text{(14)} \)

Since \( T_{s1} \) and \( T_{s2} \) would make mode non-linear, therefore, they would be calculated externally and input to model as \( T_{s1} \) and \( T_{s2} \).
### Telescope Structure
#### Altitude Model

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $k_m$, $F_0$, $L_0$, $B_m$, $B_0$, $I_n$, $I_m$
- $L_B$, $L_m$
- $L_A$, $L_m$, $L_B$

Matrix $X$ and $Y$.

---

96
Telescope Structure
Altitude Model

\[ D \]

\[ \begin{bmatrix} D & D \\ D & D \\ D & D \\ D & D \\ D & D \\ \frac{1}{L} \end{bmatrix}_{6 \times 1} \]

\[ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{J_o} & 0 \\ 0 & \frac{1}{J_m} \\ 0 & 0 \end{bmatrix}_{9 \times 2} \]

\[ \dot{x}_{q1} = A_{q1} x_{q1} + B_{q1} u_{1x1} + E_{q2} d_{3x1} \]

\[ y_{3x1} = C_{3xq} x_{q1} + D_{3x1} u_{1x1} \]
Telescope Structure
Altitude Model

\[ C \text{ matrix} \]

\[ \theta_J = \theta_J \]

\[ \omega_J = \omega_J \]

\[ x_t = \text{distance of guide star to center of close loop feedback sensor} \]

\[ x_t = L \sin \left[ 2(\theta_{jm} + \theta_{sf}) \right] + L \sin \left( \theta_{jo} + \theta_{sp} \right) \]

\[ \text{use small angle approximation} \]

\[ x_t = 2L \theta_{sf} + L \theta_{jo} + 2L \theta_{jm} \]

\[ C \text{ matrix} \]

\[ \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 3L & L & 2L & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]

for control purpose, project \( x_t \) back to rotational center as angle:

\[ \theta_t = \theta_J + \frac{x_t}{L} \]

\[ = \theta_J + 3 \theta_{sf} + \theta_{jo} + 2L \theta_{jm} \]

\[ C \text{ matrix} \]

\[ \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]
Appendix D: Matlab Simulation Model

% this is the Matlab Simulink model "telescope_model_full"
% to use it saves it as mdl file
%
%
% Chi-Yeh Hsu

Model {
    Name            "telescope_model_full"
    Version         6.1
    MdlSubVersion       0
    GraphicalInterface {
        NumRootInports      0
        NumRootOutports     0
        ParameterArgumentNames  ""
        ComputedModelVersion    "1.254"
        NumModelReferences      0
        NumTestPointedSignals   0
    }
    SavedCharacterEncoding  "ibm-1386_P100-2002"
    SaveDefaultBlockParams on
    SampleTimeColors    off
    LibraryLinkDisplay       "none"
    WideLines       off
    ShowLineDimensions off
    ShowPortDataTypes  off
    ShowLoopsOnError on
    IgnoreBidirectionalLines off
    ShowStorageClass off
    ShowTestPointIcons on
    ShowViewerIcons on
    SortedOrder         off
    ExecutionContextIcon    off
    ShowLinearizationAnnotations on
    RecordCoverage off
    CovPath             "/"
    CovSaveName         "covdata"
    CovMetricSettings   "dw"
    CovNameIncrementing off
    CovHtmlReporting on
covSaveCumulativeToWorkspaceVar on
    CovSaveSingleToWorkspaceVar on
    CovCumulativeVarName  "covCumulativeData"
    CovCumulativeReport off
    CovReportOnPause on
    ScopeRefreshTime    0.035000
    OverrideScopeRefreshTime on
    DisableAllScopes off
    DataTypeOverride    "UseLocalSettings"
    MinMaxOverflowLogging    "UseLocalSettings"
    MinMaxOverflowArchiveMode "Overwrite"
    BlockNameDataTip off
}
BlockParametersDataTip off
BlockDescriptionStringDataTip off
ToolBar on
StatusBar on
BrowserShowLibraryLinks on
BrowserLookUnderMasks on
StopFcn "Plot_FUNCTION_Azimuth\nPlot_FUNCTION_Altitude\nPlot"
"_FUNCTION_2Axis"
Created "Thu Apr 22 12:09:46 2010"
UpdateHistory "UpdateHistoryNever"
ModifiedByFormat "%<Auto>
LastModifiedBy "Chi-Yeh Hsu"
ModifiedDateFormat "%<Auto>
LastModifiedDate "Thu Aug 12 14:13:28 2010"
ModelVersionFormat "1.%<AutoIncrement:254>"
ConfigurationManager "None"
LinearizationMsg "none"
Profile off
ParamWorkspaceSource "MATLABWorkspace"
AccelSystemTargetFile "accel.tlc"
AccelTemplateMakefile "accel_default_tmf"
AccelMakeCommand "make_rtw"
TryForcingSFcnDF off
ExtModeBatchMode off
ExtModeEnableFloating on
ExtModeTrigType "manual"
ExtModeTrigMode "normal"
ExtModeTrigPort "1"
ExtModeTrigElement "any"
ExtModeTrigDuration 1000
ExtModeTrigDurationFloating "auto"
ExtModeTrigHoldOff 0
ExtModeTrigDelay 0
ExtModeTrigDirection "rising"
ExtModeTrigLevel 0
ExtModeArchiveMode "off"
ExtModeAutoIncOneShot off
ExtModeIncDirWhenArm off
ExtModeAddSuffixToVar off
ExtModeWriteAllDataToWs off
ExtModeArmWhenConnect on
ExtModeSkipDownloadWhenConnect off
ExtModeLogAll on
ExtModeAutoUpdateStatusClock on
BufferReuse on
StrictBusMsg "None"
ProdHWDeviceType "32-bit Generic"
ShowModelReferenceBlockVersion off
ShowModelReferenceBlockIO off
Array {
  Type "Handle"
  Dimension 1
  Simulink.ConfigSet {
    $ObjectID 1
    Version "1.0.4"
    Array {
      Type "Handle"
    }
Dimension  7
Simulink.SolverCC {
  $ObjectID       2
  Version         "1.0.4"
  StartTime       "0.0"
  StopTime        "t_end"
  AbsTol          "auto"
  FixedStep       "0.0001"
  InitialStep     "auto"
  MaxNumMinSteps  "-1"
  MaxOrder        5
  ExtrapolationOrder  4
  NumberNewtonIterations  1
  MaxStep         "auto"
  MinStep         "auto"
  RelTol          "1e-14"
  SolverMode      "Auto"
  Solver          "ode3"
  SolverName      "ode3"
  ZeroCrossControl "UseLocalSettings"
  PositivePriorityOrder off
  AutoInsertRateTranBlk off
  SampleTimeConstraint "Unconstrained"
  RateTranMode    "Deterministic"
}
Simulink.DataIOCC {
  $ObjectID       3
  Version         "1.0.4"
  Decimation      "1"
  ExternalInput   "[t, u]"
  FinalStateName  "xFinal"
  InitialState    "xInitial"
  LimitDataPoints on
  MaxDataPoints   "1000"
  LoadExternalInput off
  LoadInitialState off
  SaveFinalState  off
  SaveFormat      "Array"
  SaveOutput      on
  SaveState       off
  SignalLogging   on
  SaveTime        on
  StateSaveName   "xout"
  TimeSaveName    "tout"
  OutputSaveName  "yout"
  SignalLoggingName "logsout"
  OutputOption    "RefineOutputTimes"
  OutputTimes     "[]"
  Refine          "1"
}
Simulink.OptimizationCC {
  $ObjectID       4
  Array {        
    Type          "Cell"
    Dimension     5
    Cell "ZeroExternalMemoryAtStartup"
    Cell "ZeroInternalMemoryAtStartup"
}
Cell  "InitFltsAndDblsToZero"
Cell  "OptimizeModelRefInitCode"
Cell  "NoFixptDivByZeroProtection"
PropName  "DisabledProps"
}
Version  "1.0.4"
BlockReduction  on
BooleanDataType  on
ConditionallyExecuteInputs  on
ConditionalExecOptimization  "on_for_testing"
InlineParams  off
InlineInvariantSignals  on
OptimizeBlockIOStorage  on
BufferReuse  on
EnforceIntegerDowncast  on
ExpressionFolding  on
FoldNonRolledExpr  on
LocalBlockOutputs  on
ParameterPooling  on
RollThreshold  5
SystemCodeInlineAuto  off
StateBitsets  off
DataBitsets  off
UseTempVars  off
ZeroExternalMemoryAtStartup  on
ZeroInternalMemoryAtStartup  on
InitFltsAndDblsToZero  on
NoFixptDivByZeroProtection  off
OptimizeModelRefInitCode  off
LifeSpan  "inf"
}
Simulink.DebuggingCC {
$ObjectID  5
Version  "1.0.4"
RTPrefix  "error"
ConsistencyChecking  "none"
ArrayBoundsChecking  "none"
AlgebraicLoopMsg  "warning"
ArtificialAlgebraicLoopMsg  "warning"
CheckSSInitialOutputMsg  on
CheckExecutionContextPreStartOutputMsg  off
CheckExecutionContextRuntimeOutputMsg  off
SignalResolutionControl  "TryResolveAllWithWarning"
BlockPriorityViolationMsg  "warning"
MinStepSizeMsg  "warning"
SolverPrmCheckMsg  "warning"
InheritedTsInSrcMsg  "warning"
DiscreteInheritContinuousMsg  "warning"
MultiTaskDSMMsg  "warning"
MultiTaskRateTransMsg  "error"
SingleTaskRateTransMsg  "none"
TasksWithSamePriorityMsg  "warning"
CheckMatrixSingularityMsg  "none"
IntegerOverflowMsg  "warning"
Int32ToFloatConvMsg  "warning"
ParameterDowncastMsg  "error"
ParameterOverflowMsg  "error"
ModelReferenceNumInstancesAllowed "Multi"
ModelReferencePassRootInputsByReference on
ModelReferenceMinAlgLoopOccurrences off
}
Simulink.RTWCC {
$BackupClass        "Simulink.RTWCC"
$ObjectID       8
Array {
  Type            "Cell"
  Dimension           1
  Cell IncludeHyperlinkInReport "DisabledProps"
}
Version         "1.0.4"
SystemTargetFile    "grt.tlc"
GenCodeOnly       off
MakeCommand       "make_rtw"
TemplateMakefile  "grt_default_tmf"
GenerateReport     off
SaveLog           off
RTWVerbose        on
RetainRTWFile     off
ProfileTLC        off
TLCCoverage       off
TLCAssert         off
ProcessScriptMode "Default"
ConfigurationMode "Optimized"
ConfigAtBuild     off
IncludeHyperlinkInReport off
LaunchReport      off
Array {
  Type            "Handle"
  Dimension           2
  Simulink.CodeAppCC {
    $ObjectID           9
    Array {
      Type            "Cell"
      Dimension           9
      Cell IgnoreCustomStorageClasses
      Cell InsertBlockDesc
      Cell SFDataObjDesc
      Cell SimulinkDataObjDesc
      Cell DefineNamingRule
      Cell SignalNamingRule
      Cell ParamNamingRule
      Cell InlinedPrmAccess
      Cell CustomSymbolStr
      PropName        "DisabledProps"
    }
  Version         "1.0.4"
  ForceParamTrailComments off
  GenerateComments on
  IgnoreCustomStorageClasses on
  IncHierarchyInIds       off
  MaxIdLength         31
  PreserveName       off
  104

PreserveNameWithParent off
ShowEliminatedStatement off
IncAutoGenComments off
SimulinkDataObjDesc off
SPDataObjDesc off
IncDataTypeInIds off
PrefixModelToSubsysFcnNames on
CustomSymbolStr "$R$N$M"
MangleLength 1
DefineNamingRule "None"
ParamNamingRule "None"
SignalNamingRule "None"
InsertBlockDesc off
SimulinkBlockComments on
EnableCustomComments off
InlinedPmAccess "Literals"
ReqsInCode off

Simulink.GRTTargetCC {
$BackupClass "Simulink.TargetCC"
$ObjectID 10
Array {
Type "Cell"
Dimension 12
Cell "IncludeMdlTerminateFcn"
Cell "CombineOutputUpdateFcns"
Cell "SuppressErrorStatus"
Cell "ERTCustomFileBanners"
Cell "GenerateSampleERTMain"
Cell "MultiInstanceERTCode"
Cell "PurelyIntegerCode"
Cell "SupportNonFinite"
Cell "SupportComplex"
Cell "SupportAbsoluteTime"
Cell "SupportContinuousTime"
Cell "SupportNonInlinedSFcns"
PropName "DisabledProps"
}
Version "1.0.4"
TargetFcnLib "ansi_tfl_tmw.mat"
GenFloatMathFcnCalls "ANSI_C"
UtilityFuncGeneration "Auto"
GenerateFullHeader on
GenerateSampleERTMain off
IsPIITarget off
ModelReferenceCompliant on
IncludeMdlTerminateFcn on
CombineOutputUpdateFcns off
SuppressErrorStatus off
IncludeFileDelimiter "Auto"
ERTCustomFileBanners off
SupportAbsoluteTime on
LogVarNameModifier "rt_"
MatFileLogging on
MultiInstanceERTCode off
SupportNonFinite on
SupportComplex on
 PurelyIntegerCode off
SupportContinuousTime on
SupportNonInlinedFunctions on
ExtMode off
ExtModeStaticAllocation off
ExtModeTesting off
ExtModeStaticAllocationSize 1000000
ExtModeTransport 0
ExtModeMexFile "ext_comm"
RTWCAPISignals off
RTWCAPIParameters off
RTWCAPISettings off
GenerateASAP2 off
}
PropName "Components"
}
PropName "Components"
}
Name "Configuration"
SimulationMode "normal"
CurrentDlgPage "Solver"
}
PropName "ConfigurationSets"
}
Simulink.ConfigSet {
$PropName "ActiveConfigurationSet"
$ObjectID 1
}
BlockDefaults {
Orientation "right"
ForegroundColor "black"
BackgroundColor "white"
DropShadow off
NamePlacement "normal"
FontName "Helvetica"
FontSize 10
FontWeight "normal"
FontAngle "normal"
ShowName on
}
BlockParameterDefaults {
Block {
BlockType Clock
DisplayTime off
}
Block {
BlockType Constant
Value "1"
VectorParameters on
OutDataTypeMode "Inherit from 'Constant value'"
OutDataType "sfix(16)"
ConRadixGroup "Use specified scaling"
OutScaling "2^0"
SampleTime "inf"
}
Block {

BlockType Demux
Outputs "4"
DisplayOption "none"
BusSelectionMode off
}
Block {
    BlockType Derivative
    LinearizePole "inf"
}
Block {
    BlockType DiscretePulseGenerator
    PulseType "Sample based"
    TimeSource "Use simulation time"
    Amplitude "1"
    Period "2"
    PulseWidth "1"
    PhaseDelay "0"
    SampleTime "1"
    VectorParams1D on
}
Block {
    BlockType Gain
    Gain "1"
    Multiplication "Element-wise(K.*u)"
    ParameterDataTypeMode "Same as input"
    ParameterDataType "sfix(16)"
    ParameterScalingMode "Best Precision: Matrix-wise"
    ParameterScaling "2^0"
    OutDataTypeMode "Same as input"
    OutDataType "sfix(16)"
    OutScaling "2^0"
    LockScale off
    RndMeth "Floor"
    SaturateOnIntegerOverflow on
    SampleTime "-1"
}
Block {
    BlockType SignalConversion
    OverrideOpt off
}
Block {
    BlockType Inport
    BusObject "BusObject"
    BusOutputAsStruct off
    PortDimensions "-1"
    SampleTime "-1"
    DataType "auto"
    OutDataType "sfix(16)"
    OutScaling "2^0"
    SignalType "auto"
    SamplingMode "auto"
    Interpolate on
}
Block {
    BlockType Integrator
    ExternalReset "none"
    InitialConditionSource "internal"
InitialCondition        "0"
LimitOutput         off
UpperSaturationLimit    "inf"
LowerSaturationLimit    "-inf"
ShowSaturationPort      off
ShowStatePort       off
AbsoluteTolerance       "auto"
IgnoreLimit         off
ZeroCross           on
}
Block {
    BlockType           Math
    Operator            "exp"
    OutputSignalType        "auto"
    SampleTime          "-1"
    OutDataTypeMode   "Same as first input"
    OutDataType        "sfix(16)"
    OutScaling          "2^0"
    LockScale           off
    RndMeth             "Floor"
    SaturateOnIntegerOverflow on
}
Block {
    BlockType           Mux
    Inputs              "4"
    DisplayOption       "none"
    BusObject           "BusObject"
    NonVirtualBus       off
}
Block {
    BlockType           Outport
    Port            "1"
    BusObject           "BusObject"
    BusOutputAsStruct       off
    PortDimensions          "-1"
    SampleTime          "-1"
    DataType            "auto"
    OutDataType        "sfix(16)"
    OutScaling          "2^0"
    SignalType          "auto"
    SamplingMode        "auto"
    OutputWhenDisabled      "held"
    InitialOutput       "[]"
}
Block {
    BlockType           Product
    Inputs              "2"
    Multiplication        "Element-wise(.)"
    InputSameDT         on
    OutDataTypeMode   "Same as first input"
    OutDataType        "sfix(16)"
    OutScaling          "2^0"
    LockScale           off
    RndMeth             "Zero"
    SaturateOnIntegerOverflow on
    SampleTime          "-1"
}
Block {
    BlockType        RateTransition
    Integrity        on
    Deterministic   on
    X0              "0"
    OutPortSampleTime    "-1"
}

Block {
    BlockType        Saturate
    UpperLimit      "0.5"
    LowerLimit      "-0.5"
    LinearizeAsGain on
    ZeroCross       on
    SampleTime      "-1"
}

Block {
    BlockType        Scope
    Floating        off
    ModelBased      off
    TickLabels      "OneTimeTick"
    ZoomMode        "on"
    Grid            "on"
    TimeRange       "auto"
    YMin            "-5"
    YMax            "5"
    SaveToWorkspace off
    SaveName        "ScopeData"
    LimitDataPoints on
    MaxDataPoints   "5000"
    Decimation      "1"
    SampleInput     off
    SampleTime      "0"
}

Block {
    BlockType        "S-Function"
    FunctionName     "system"
    SFunctionModules "[]"
    PortCounts       "[]"
}

Block {
    BlockType        Signum
    ZeroCross        on
    SampleTime      "-1"
}

Block {
    BlockType        Step
    Time             "1"
    Before           "0"
    After            "1"
    SampleTime      "-1"
    VectorParams1D   on
    ZeroCross        on
}

Block {
    BlockType        SubSystem
    ShowPortLabels   on
    Permissions      "ReadWrite"
PermitHierarchicalResolution "All"
SystemSampleTime "-1"
RTWFcnNameOpts "Auto"
RTWFileNameOpts "Auto"
SimViewingDevice off
DataTypeOverride "UseLocalSettings"
MinMaxOverflowLogging "UseLocalSettings"

Block {
  BlockType Sum
  IconShape "rectangular"
  Inputs "++"
  InputSameDT on
  OutDataTypeMode "Same as first input"
  OutDataType "sfix(16)"
  OutScaling "2^0"
  LockScale off
  RndMeth "Floor"
  SaturateOnIntegerOverflow on
  SampleTime "-1"
}
Block {
  BlockType Switch
  Criteria "u2 >= Threshold"
  Threshold "0"
  InputSameDT on
  OutDataTypeMode "Inherit via internal rule"
  RndMeth "Floor"
  SaturateOnIntegerOverflow on
  ZeroCross on
  SampleTime "-1"
}
Block {
  BlockType ToWorkspace
  VariableName "simulink_output"
  MaxDataPoints "1000"
  Decimation "1"
  SampleTime "0"
  FixptAsFi off
}
Block {
  BlockType Trigonometry
  Operator "sin"
  OutputSignalType "auto"
  SampleTime "-1"
}
Block {
  BlockType UniformRandomNumber
  Minimum "-1"
  Maximum "1"
  Seed "0"
  SampleTime "-1"
  VectorParams1D on
}
AnnotationDefaults {
  HorizontalAlignment "center"
}
VerticalAlignment "middle"
ForegroundColor "black"
BackgroundColor "white"
DropShadow off
FontName "Helvetica"
FontSize 10
FontWeight "normal"
FontAngle "normal"

LineDefaults {
  FontName "Helvetica"
  FontSize 9
  FontWeight "normal"
  FontAngle "normal"
}

System {
  Name "telescope_model_full"
  Location [216, 72, 1438, 849]
  Open on
  ModelBrowserVisibility on
  ModelBrowserWidth 214
  ScreenColor "white"
  PaperOrientation "landscape"
  PaperPositionMode "auto"
  PaperType "usletter"
  PaperUnits "inches"
  ZoomFactor "100"
  ReportName "simulink-default.rpt"
  Block {
    BlockType Constant
    Name "0Input"
    Position [405, 155, 435, 185]
    Value "0"
  }
  Block {
    BlockType Constant
    Name "0Input1"
    Position [25, 295, 55, 325]
    Value "pi/12"
  }
  Block {
    BlockType Constant
    Name "0Input2"
    Position [385, 430, 415, 460]
    Value "0"
  }
  Block {
    BlockType Constant
    Name "0Input3"
    Position [20, 50, 50, 80]
    Value "pi/12"
  }
  Block {
    BlockType Reference
    Name "Altitude\nInput\nSwitch"
    Ports [2, 1]
    Position [440, 347, 470, 383]
Name            "Feedback"
Position        [40, 208, 70, 222]
Port            "2"
IconDisplay      "Port number"
LatchInput      off

Block {
    BlockType       Sum
    Name            "Add"
    Ports           [3, 1]
    Position        [655, 107, 685, 173]
    Inputs          "+++"
    InputSameDT     off
    OutDataTypeMode  "Inherit via internal rule"
    OutScaling      "2^-10"
    SaturateOnIntegerOverflow off
}

Block {
    BlockType       Scope
    Name            "Control Signals"
    Ports           [4]
    Position        [830, 194, 880, 316]
    Location        [1, 46, 1441, 869]
    Open            off
    NumInputPorts   "4"
    List {
        ListType            AxesTitles
        axes1           "%<SignalLabel>"
        axes2           "%<SignalLabel>"
        axes3           "%<SignalLabel>"
        axes4           "%<SignalLabel>"
    }
    YMin            "-5~-5~-5~-5"
    YMax            "5~5~5~5"
    SaveName        "ScopeData1"
    DataFormat      "StructureWithTime"
    LimitDataPoints off
}

Block {
    BlockType       Derivative
    Name            "Derivative"
    Position        [380, 125, 410, 155]
}

Block {
    BlockType       Gain
    Name            "Derivative_Gain"
    Position        [290, 123, 335, 157]
    Gain            "Kd_a"
    ParameterDataTypeMode  "Inherit via internal rule"
    ParameterDataType   "fixdt(1, 16)"
    OutDataTypeMode      "Inherit via internal rule"
    OutDataType          "fixdt(1, 16)"
    SaturateOnIntegerOverflow off
}

Block {
    BlockType       Gain
    Name            "Gain"
Position [855, 133, 885, 147]
IconDisplay "Port number"
BusOutputAsStruct off

Line {
  SrcBlock "Integral_Gain"
  SrcPort 1
  DstBlock "Integrator"
  DstPort 1
}
Line {
  SrcBlock "Derivative_Gain"
  SrcPort 1
  DstBlock "Derivative"
  DstPort 1
}
Line {
  SrcBlock "Proportional_Gain"
  SrcPort 1
  Points [220, 0]
  Branch {
    Points [20, 0; 0, -45]
    DstBlock "Add"
    DstPort 3
  }
  Branch {
    Points [0, 95]
    DstBlock "Control Signals"
    DstPort 4
  }
}
Line {
  SrcBlock "Sum"
  SrcPort 1
  Points [15, 0]
  Branch {
    DstBlock "Derivative_Gain"
    DstPort 1
  }
  Branch {
    Points [0, -65]
    DstBlock "Proportional_Gain"
OutScaling          "2^-10"
SaturateOnIntegerOverflow off
Port {
    PortNumber          1
    Name            "dX/dt"
    RTWStorageClass    "Auto"
    DataLoggingNameMode "SignalName"
    ShowSigGenPortName on
}
}
Block {
    BlockType       Sum
    Name            "Add1"
    Ports           [2, 1]
    Position        [590, 267, 620, 298]
    InputSameDT      off
    OutDataTypeMode "Inherit via internal rule"
    OutScaling          "2^-10"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType       Gain
    Name            "C"
    Position        [485, 253, 545, 297]
    Gain            "C_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType       Gain
    Name            "D"
    Position        [340, 418, 400, 472]
    Gain            "D_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
    OutDataTypeMode "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType       Demux
    Name            "Demux"
    Ports           [1, 3]
    Position        [690, 230, 695, 340]
    BackgroundColor "black"
    ShowName        off
    Outputs         "3"
    DisplayOption   "bar"
}
Block {
    BlockType       Gain
    Name            "Disturbance_Gain_E"
    Position        [175, 117, 225, 163]
    Gain            "E_a"
    Multiplication  "Matrix(K*u)"
    ParameterDataTypeMode "Inherit via internal rule"
OutDataTypeMode     "Inherit via internal rule"
SaturateOnIntegerOverflow off
}

Block {
    BlockType       Gain
    Name            "Input_Gain_B"
    Position        [180, 252, 230, 298]
    Gain            "B_a"
    Multiplication   "Matrix(K*u)"
    ParameterDataTypeMode   "Inherit via internal rule"
    OutDataTypeMode     "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}

Block {
    BlockType       Integrator
    Name            "Integrator"
    Ports           [1, 1]
    Position        [375, 260, 405, 290]
    InitialCondition    "X_a_ini"
    IgnoreLimit         off
    Port {
        PortNumber          1
        Name            "X"
        RTWStorageClass   "Auto"
        DataLoggingNameMode     "SignalName"
        ShowSigGenPortName      on
    }
}

Block {
    BlockType       Gain
    Name            "System_Gain_A"
    Position        [335, 334, 395, 386]
    Orientation     "left"
    Gain            "A_a"
    Multiplication   "Matrix(K*u)"
    ParameterDataTypeMode   "Inherit via internal rule"
    OutDataTypeMode     "Inherit via internal rule"
    SaturateOnIntegerOverflow off
}

Block {
    BlockType       ToWorkspace
    Name            "To Workspace"
    Position        [555, 155, 670, 185]
    VariableName    "Altitude_System_State"
    MaxDataPoints   "inf"
    SampleTime      "-1"
    SaveFormat      "Array"
}

Block {
    BlockType       Outport
    Name            "Theta(Theta_J_a)"
    Position        [785, 243, 815, 257]
    IconDisplay     "Port number"
    BusOutputAsStruct   off
}

Block {
    BlockType       Outport
    Name            ""
Name            "Omega(Theta_dot_J_a)"
Position        [785, 278, 815, 292]
Port            "2"
IconDisplay     "Port number"
BusOutputAsStruct   off
}
Block {
  BlockType       Outport
  Name            "X_alt(m)"
  Position        [785, 313, 815, 327]
  Port            "3"
  IconDisplay     "Port number"
  BusOutputAsStruct   off
}
Line {
  Name            "X"
  Labels          [0, 1]
  SrcBlock        "Integrator"
  SrcPort         1
  Points          [30, 0]
  Branch {
    DstBlock            "C"
    DstPort         1
  }
  Branch {
    Points          [0, -105]
    DstBlock            "To Workspace"
    DstPort         1
  }
  Branch {
    Points          [0, 85]
    DstBlock            "System_Gain_A"
    DstPort         1
  }
}
Line {
  Name            "dX/dt"
  Labels          [0, 1]
  SrcBlock        "Add"
  SrcPort         1
  DstBlock        "Integrator"
  DstPort         1
}
Line {
  SrcBlock        "Input_Gain_B"
  SrcPort         1
  DstBlock        "Add"
  DstPort         2
}
Line {
  SrcBlock        "C"
  SrcPort         1
  DstBlock        "Add1"
  DstPort         1
}
Line {
  SrcBlock        "D"
Position [160, 93, 325, 137]
TreatAsAtomicUnit off
MinAlgLoopOccurrences off
RTWSystemCode "Auto"

System {
  Name "Azimuth_PID_Controller"
  Location [251, 138, 1179, 773]
  Open off
  ModelBrowserVisibility on
  ModelBrowserWidth 200
  ScreenColor "white"
  PaperOrientation "landscape"
  PaperPositionMode "auto"
  PaperType "usletter"
  PaperUnits "inches"
  ZoomFactor "100"

Block {
  BlockType Inport
  Name "Control_Input"
  Position [40, 133, 70, 147]
  Port "1"
  IconDisplay "Port number"
  LatchInput off
}

Block {
  BlockType Inport
  Name "Feedback"
  Position [40, 208, 70, 222]
  Port "2"
  IconDisplay "Port number"
  LatchInput off
}

Block {
  BlockType Sum
  Name "Add"
  Ports [3, 1]
  Position [655, 107, 685, 173]
  Inputs "+++"
  InputSameDT off
  OutDataTypeMode "Inherit via internal rule"
  OutScaling "2^-10"
  SaturateOnIntegerOverflow off
}

Block {
  BlockType Scope
  Name "Control Signals"
  Ports [4]
  Position [830, 194, 880, 316]
  Location [5, 54, 1445, 877]
  Open off
  NumInputPorts "4"
  List {
    ListType AxesTitles
    axes1 "%%<SignalLabel>"
    axes2 "%%<SignalLabel>"
    axes3 "%%<SignalLabel>"
    axes4 "%%<SignalLabel>"
  }
}
Line {
    SrcBlock        "Proportional_Gain"
    SrcPort         1
    Points          [240, 0]
    Branch {
        Points          [50, 0; 0, 45]
        DstBlock            "Add"
        DstPort         1
    }
    Branch {
        Points          [0, 165]
        DstBlock            "Control Signals"
        DstPort         2
    }
}

Line {
    SrcBlock        "Integrator"
    SrcPort         1
    Points          [180, 0]
    Branch {
        Points          [20, 0; 0, -45]
        DstBlock            "Add"
        DstPort         3
    }
    Branch {
        Points          [0, 95]
        DstBlock            "Control Signals"
        DstPort         4
    }
}

Line {
    SrcBlock        "Sum"
    SrcPort         1
    Points          [5, 0]
    Branch {
        Points          [10, 0]
        Branch {
            DstBlock            "Derivative_Gain"
            DstPort             1
        }
        Branch {
            Points          [0, -65]
            DstBlock            "Proportional_Gain"
            DstPort             1
        }
        Branch {
            Points          [0, 65]
            DstBlock            "Integral_Gain"
            DstPort             1
        }
        Branch {
            Points          [0, 145]
            DstBlock            "Scope"
            DstPort             1
        }
    }
}
Multiplication "Matrix(K*u)"
ParameterDataTypeMode "Inherit via internal rule"
OutDataTypeMode "Inherit via internal rule"
SaturateOnIntegerOverflow off

Block {
BlockType Integrator
Name "Integrator"
Ports [1, 1]
Position [375, 260, 405, 290]
InitialCondition "X_ini"
IgnoreLimit off
Port {
  PortNumber 1
  Name "X"
  RTWStorageClass "Auto"
  DataLoggingNameMode "SignalName"
  ShowSigGenPortName on
}

Block {
BlockType Gain
Name "System_Gain_A"
Position [335, 334, 395, 386]
Orientation "left"
Gain "A"
Multiplication "Matrix(K*u)"
ParameterDataTypeMode "Inherit via internal rule"
OutDataTypeMode "Inherit via internal rule"
SaturateOnIntegerOverflow off
}

Block {
BlockType ToWorkspace
Name "To Workspace"
Position [555, 155, 670, 185]
VariableName "Azimuth_System_State"
MaxDataPoints "inf"
SampleTime "-1"
SaveFormat "Array"
}

Block {
BlockType Outport
Name "Theta(Theta_J)"
Position [785, 243, 815, 257]
IconDisplay "Port number"
BusOutputAsStruct off
}

Block {
BlockType Outport
Name "Omega(theta_dot_J)"
Position [785, 278, 815, 292]
Port "2"
IconDisplay "Port number"
BusOutputAsStruct off
}

Block {
BlockType Outport

DstBlock        "Demux"
DstPort         1
}
Line {  
SrcBlock       "Control_Input(V)"
SrcPort         1
Points          [35, 0]
Branch {  
  DstBlock        "Input_Gain_B"
  DstPort         1
  }
Branch {  
  Points          [0, 170]
  DstBlock        "D"
  DstPort         1
  }
}
Line {  
SrcBlock       "Demux"
SrcPort         1
DstBlock        "Theta(Theta_J)"
DstPort         1
}
Line {  
SrcBlock       "Demux"
SrcPort         2
DstBlock        "Omega(theta_dot_J)"
DstPort         1
}
Line {  
SrcBlock       "Demux"
SrcPort         3
DstBlock        "X_azi(m)"
DstPort         1
}
Line {  
SrcBlock       "Disturbance(Ts)"
SrcPort         1
DstBlock        "Disturbance_Gain_E"
DstPort         1
}
Line {  
SrcBlock       "System_Gain_A"
SrcPort         1
Points          [-60, 0; 0, -60]
DstBlock        "Add"
DstPort         3
}
Line {  
SrcBlock       "Disturbance_Gain_E"
SrcPort         1
Points          [25, 0; 0, 110]
DstBlock        "Add"
DstPort         1
}
Annotation {  
Name            "Telescope Azimuth Structure Model"
Name: "Azimuth_Angular_Position\n(Theta_J)(radius)"
Position: [90, 703, 120, 717]
Port: "2"
IconDisplay: "Port number"
LatchInput: off

Block {
BlockType: Inport
Name: "Wind_Direction\n(radius)"
Position: [115, 353, 145, 367]
Port: "3"
IconDisplay: "Port number"
LatchInput: off
}

Block {
BlockType: Inport
Name: "Wind_Speed\n(m/s)"
Position: [115, 423, 145, 437]
Port: "4"
IconDisplay: "Port number"
LatchInput: off
}

Block {
BlockType: SubSystem
Name: "Altitude_Wind_Model"
Ports: [5, 3]
Position: [400, 112, 770, 468]
TreatAsAtomicUnit: off
MinAlgLoopOccurrences: off
RTWSYSTEMCode: "Auto"
System {
Name: "Altitude_Wind_Model"
Location: [209, 80, 1419, 701]
Open: off
ModelBrowserVisibility: on
ModelBrowserWidth: 200
ScreenColor: "white"
PaperOrientation: "landscape"
PaperPositionMode: "auto"
PaperType: "usletter"
PaperUnits: "inches"
ZoomFactor: "100"
Block {
BlockType: Inport
Name: "Telescope_Azimuth\nAngular_Position\n(Theta_J_a)(radius)"
Position: [80, 133, 110, 147]
Port: "1"
IconDisplay: "Port number"
LatchInput: off
}
Block {
BlockType: Inport
Name: "OTA_Altitude_Angular_Position\n(Theta_J_O_a)(radius)"
Position: [80, 208, 110, 222]
Port: "2"
BlockType           Product
Name            "Product3"
Ports           [2, 1]
Position       [605, 137, 635, 168]
InputSameDT    off
OutDataTypeMode "Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
  BlockType           Product
  Name            "Product4"
  Ports           [2, 1]
  Position       [615, 292, 645, 323]
  InputSameDT    off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType           Product
  Name            "Product5"
  Ports           [2, 1]
  Position       [615, 422, 645, 453]
  InputSameDT    off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType           Signum
  Name            "Sign"
  Position       [510, 75, 540, 105]
}
Block {
  BlockType           Signum
  Name            "Sign1"
  Position       [515, 230, 545, 260]
}
Block {
  BlockType           Signum
  Name            "Sign2"
  Position       [520, 365, 550, 395]
}
Block {
  BlockType           Trigonometry
  Name            "Trigonometric\nFunction"
  Ports           [1, 1]
  Position       [275, 135, 305, 165]
  Operator        "cos"
}
Block {
  BlockType           Trigonometry
  Name            "Trigonometric\nFunction1"
  Ports           [1, 1]
  Position       [275, 300, 305, 330]
}
Block {
  BlockType           Trigonometry
  Name            "Trigonometric\nFunction2"
Line {
    SrcBlock            "Gain4"
    SrcPort             1
    DstBlock            "Wind_Torque_Fork
                        (Ts3_a)(Nm)"
    DstPort             1
}
Line {
    SrcBlock            "OTA_Altitude_Angular_Position
                        (Theta_J
                  _O_a)(radius)"
    SrcPort             1
    DstBlock            "Trigonometric
                        nFunction2"
    DstPort             1
}
Line {
    SrcBlock            "Trigonometric
                        nFunction1"
    SrcPort             1
    DstBlock            "Product1"
    DstPort             2
}
Line {
    SrcBlock            "mm_Altitude_Angular_Position
                        (Theta_Jm
                   _O_a)(radius)"
    SrcPort             1
    DstBlock            "Trigonometric
                        nFunction1"
    DstPort             1
}
Line {
    SrcBlock            "Trigonometric
                        nFunction2"
    SrcPort             1
    Points              [55, 0; 0, -55]
    DstBlock            "Product"
    DstPort             2
}
Line {
    SrcBlock            "Trigonometric
                        nFunction"
    SrcPort             1
    Points              [0, 0; 25, 0]
    Branch {
        Points           [0, 285]
        DstBlock         "Product2"
        DstPort          1
    }
    Branch {
        Points           [15, 0]
        Branch {
            Points           [0, 155]
            DstBlock         "Product1"
            DstPort          1
        }
        Branch {
            DstBlock         "Product"
            DstPort          1
        }
    }
}
Branch {
  DstBlock        "Math\nFunction1"
  DstPort         1
}

Branch {
  Points          [0, -70]
  DstBlock        "Sign1"
  DstPort         1
}

Line {
  SrcBlock            "Math\nFunction1"
  SrcPort             1
  DstBlock            "Product4"
  DstPort             2
}

Line {
  SrcBlock            "Sign1"
  SrcPort             1
  Points              [25, 0; 0, 55]
  DstBlock            "Product4"
  DstPort             1
}

Line {
  SrcBlock            "Product4"
  SrcPort             1
  DstBlock            "Gain3"
  DstPort             1
}

Line {
  SrcBlock            "Product2"
  SrcPort             1
  Points              [0, 0; 35, 0]
  Branch {
    DstBlock        "Math\nFunction2"
    DstPort         1
  }
  Points          [0, -65]
  DstBlock        "Sign2"
  DstPort         1
}

Line {
  SrcBlock            "Math\nFunction2"
  SrcPort             1
  DstBlock            "Product5"
  DstPort             2
}

Line {
  SrcBlock            "Sign2"
  SrcPort             1
  Points              [20, 0; 0, 50]
  DstBlock            "Product5"
  DstPort             1
}

Line {
BlockType: Inport
Name: "Wind_Direction\n(Theta_w)(radius)"
Position: [85, 293, 115, 307]
Port: "3"
IconDisplay: "Port number"
LatchInput: off

Block Type: Inport
Name: "Wind_Speed\n(m/s)"
Position: [85, 348, 115, 362]
Port: "4"
IconDisplay: "Port number"
LatchInput: off

Block Type: Sum
Name: "Add"
Ports: [2, 1]
Position: [210, 107, 240, 138]
Inputs: "+-"
InputSameDT: off
OutDataTypeMode: "Inherit via internal rule"
OutScaling: "2^-10"
SaturateOnIntegerOverflow: off

Block Type: Sum
Name: "Add1"
Ports: [2, 1]
Position: [210, 202, 240, 233]
Inputs: "+-"
InputSameDT: off
OutDataTypeMode: "Inherit via internal rule"
OutScaling: "2^-10"
SaturateOnIntegerOverflow: off

Block Type: Gain
Name: "Gain2"
Position: [745, 102, 985, 158]
Gain: "0.5*p_air*A_OTA*0.5*L_OTA"
ParameterDataTypeMode: "Inherit via internal rule"
SaturateOnIntegerOverflow: off

Block Type: Gain
Name: "Gain3"
Position: [745, 197, 985, 253]
Gain: "-0.5*p_air*A_mm*0.5*L_mm"
ParameterDataTypeMode: "Inherit via internal rule"
SaturateOnIntegerOverflow: off

Block Type: Math
Name: "Math\nFunction"
Ports: [1, 1]
Block {
  BlockType Math
  Name "Math\nFunction1"
  Ports [1, 1]
  Position [480, 120, 510, 150]
  Operator "square"
}
Block {
  BlockType Product
  Name "Product"
  Ports [2, 1]
  Position [390, 116, 420, 149]
  InputSameDT off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType Product
  Name "Product1"
  Ports [2, 1]
  Position [390, 211, 420, 244]
  InputSameDT off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType Product
  Name "Product2"
  Ports [2, 1]
  Position [570, 112, 600, 143]
  InputSameDT off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType Product
  Name "Product3"
  Ports [2, 1]
  Position [660, 207, 690, 238]
  InputSameDT off
  OutDataTypeMode "Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType Signum
  Name "Sign"
  Position [485, 55, 515, 85]
}
Block {
  BlockType Signum
  Name "Sign1"
  Position [540, 165, 570, 195]
}
Block {
  BlockType Math
  Name "Math\nFunction1"
  Ports [1, 1]
  Position [520, 215, 550, 245]
  Operator "square"
}
Name            "mm --> Mirror Mount"
Position        [274, 48]}
}
Annotation {
Name            "Telescope Azimuth Wind Model"
Position        [341, 24]
FontName        "Arial"
FontSize        16
FontWeight      "bold"
}
}
}
Block {
BlockType       SubSystem
Name            "Gravity_Model"
Ports           [2, 2]
Position        [400, 491, 845, 624]
TreatAsAtomicUnit   off
MinAlgLoopOccurrences   off
RTWSystemCode       "Auto"
System {
Name            "Gravity_Model"
Location        [695, 84, 1374, 426]
Open            off
ModelBrowserVisibility  on
ModelBrowserWidth  200
ScreenColor      "white"
PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType       "usletter"
PaperUnits      "inches"
ZoomFactor      "100"
Block {
BlockType       Inport
Name            "OTA_Altitude_Angular_Position
(Theta_J"
Position        [110, 103, 140, 117]
Port            "1"
IconDisplay     "Port number"
LatchInput      off
}
Block {
BlockType       Inport
Name            "mm_Altitude_Angular_Position
(Theta_Jm"
Position        [110, 238, 140, 252]
Port            "2"
IconDisplay     "Port number"
LatchInput      off
}
Block {
BlockType       Gain
Name            "Gain2"
Position        [295, 89, 480, 131]
Gain            "-mOTA*7*0.5*L OTA"
ParameterDataTypeMode   "Inherit via internal rule"
OutDataTypeMode        "Inherit via internal rule"
BusOutputAsStruct off
}
Block {
  BlockType       Outport
  Name            "Ts_a"
  Position        [995, 333, 1025, 347]
  Port            "2"
  IconDisplay     "Port number"
  BusOutputAsStruct off
}
Line {
  SrcBlock        "Altitude_Angular_Position(Theta_J_a)(radius)"
  SrcPort         1
  Points          [75, 0]
  Branch {
    Points          [130, 0]
    Branch {
      Points            [0, 375]
      DstBlock           "Gravity_Model"
      DstPort            1
    }
    Branch {
      Points            [20, 0]
      Branch {
        DstBlock           "Altitude_Wind_Model"
        DstPort            1
      }
      Branch {
        Points            [0, 70]
        DstBlock           "Altitude_Wind_Model"
        DstPort            2
      }
    }
  }
  Branch {
    Points          [0, 140; 105, 0]
    Branch {
      DstBlock           "Altitude_Wind_Model"
      DstPort            3
    }
    Branch {
      Points            [0, 300]
      DstBlock           "Gravity_Model"
      DstPort            2
    }
  }
}
Line {
  SrcBlock        "Azimuth_Angular_Position(Theta_J)(radius)"
  SrcPort         1
  Points          [65, 0]
  Branch {
    DstBlock           "Azimuth_Wind_Model"
    DstPort            1
  }
  Branch {

Block {
  BlockType       ToWorkspace
  Name            "To Workspace"
  Position        [970, 16, 1105, 54]
  VariableName    "Azimuth_Control_Input"
  MaxDataPoints   "inf"
  SampleTime      "-1"
  SaveFormat      "Array"
}

Block {
  BlockType       ToWorkspace
  Name            "To Workspace1"
  Position        [980, 261, 1115, 299]
  VariableName    "Altitude_Control_Input"
  MaxDataPoints   "inf"
  SampleTime      "-1"
  SaveFormat      "Array"
}

Block {
  BlockType       ToWorkspace
  Name            "To Workspace2"
  Position        [215, 550, 310, 600]
  Orientation     "left"
  VariableName    "Disturbances"
  MaxDataPoints   "inf"
  SampleTime      "-1"
  SaveFormat      "Array"
}

Block {
  BlockType       Constant
  Name            "Wind Direction(n(radius))"
  Position        [85, 740, 115, 770]
  Value           "0"
}

Block {
  BlockType       SubSystem
  Name            "Wind Speed\nGenerator"
  Ports           [0, 1]
  Position        [55, 648, 140, 682]
  TreatAsAtomicUnit off
  MinAlgLoopOccurrences off
  RTWSYSTEMCode    "Auto"
  System {
    Name            "Wind Speed\nGenerator"
    Location        [202, 80, 1410, 857]
    Open            off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor      "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType       "usletter"
    PaperUnits      "inches"
    ZoomFactor      "100"
  }
}

Block {
  BlockType       Sum
  Name            "Add"
}
Ports [2, 1]
Position [365, 77, 395, 108]
InputSameDT off
OutDataTypeMode "Inherit via internal rule"
OutScaling "2^-10"
SaturateOnIntegerOverflow off
}
Block {
  BlockType DiscretePulseGenerator
  Name "Pulse\nGenerator"
  Ports [0, 1]
  Position [140, 163, 185, 197]
  PulseType "Time based"
  Amplitude "Wind_amplitude"
  Period "Wind_period"
  PulseWidth "80"
  PhaseDelay "Wind_delay"
}
Block {
  BlockType RateTransition
  Name "Rate Transition"
  Position [250, 64, 290, 106]
}
Block {
  BlockType Scope
  Name "Scope1"
  Ports [1]
  Position [660, 79, 690, 111]
  Location [17, 143, 864, 783]
  Open off
  NumInputPorts "1"
  List {
    ListType AxesTitles
    axes1 "%<SignalLabel>"
  }
  TimeRange "10"
  YMin ",-2"
  SaveName "ScopeData1"
  DataFormat "StructureWithTime"
  LimitDataPoints off
}
Block {
  BlockType UniformRandomNumber
  Name "Uniform Random\nNumber"
  Position [55, 69, 85, 101]
  SampleTime "0.1"
}
Block {
  BlockType Outport
  Name "Wind_Speed\n(m/s)"
  Position [660, 18, 690, 32]
  IconDisplay "Port number"
  BusOutputAsStruct off
}
Line {
  SrcBlock "Uniform Random\nNumber"
  SrcPort 1
\% Azimuth Structure Matrix
\% System Matrix
\% A = zeros(9,9);
\% A(1,1) = 0;
\% A(1,2) = 0;
\% A(1,3) = 0;
\% A(1,4) = 0;
\% A(1,5) = 1;
\% A(1,6) = 0;
\% A(1,7) = 0;
\% A(1,8) = 0;
\% A(1,9) = 0;

A(2,1) = 0;
A(2,2) = 0;
A(2,3) = 0;
A(2,4) = 0;
A(2,5) = 0;
A(2,6) = 1;
A(2,7) = 0;
A(2,8) = 0;
A(2,9) = 0;

A(3,1) = 0;
A(3,2) = 0;
A(3,3) = 0;
A(3,4) = 0;
A(3,5) = 0;
A(3,6) = 0;
A(3,7) = 1;
A(3,8) = 0;
A(3,9) = 0;

A(4,1) = 0;
A(4,2) = 0;
A(4,3) = 0;
A(4,4) = 0;
A(4,5) = 0;
A(4,6) = 0;
A(4,7) = 0;
A(4,8) = 1;
A(4,9) = 0;

A(5,1) = -(K_f)/J_mo;
A(5,2) = (K_f)/J_mo;
A(5,3) = 0;
A(5,4) = 0;
A(5,5) = -(B_mo+B_f)/J_mo;
A(5,6) = B_f/J_mo;
A(5,7) = 0;
A(5,8) = 0;
A(5,9) = 1/(Ka*J_mo);

A(6,1) = K_f/J_am;
A(6,2) = -(K_f+K_o+K_m)/J_am;
A(6,3) = K_o/J_am;
A(6,4) = K_m/J_am;
A(6,5) = B_f/J_am;
A(6,6) = -(B_f+B_o+B_m)/J_am;
A(6,7) = B_o/J_am;
A(6,8) = B_m/J_am;
A(6,9) = 0;

A(7,1) = 0;
A(7,2) = K_o/J_o;
A(7,3) = -K_o/J_o;
A(7,4) = 0;
A(7,5) = 0;
A(7,6) = B_m/J_o;
A(7,7) = -B_o/J_o;
A(7,8) = 0;
A(7,9) = 0;

A(8,1) = 0;
A(8,2) = K_m/J_m;
A(8,3) = 0;
A(8,4) = -K_m/J_m;
A(8,5) = 0;
A(8,6) = B_m/J_m;
A(8,7) = 0;
A(8,8) = -B_m/J_m;
A(8,9) = 0;

A(9,1) = 0;
A(9,2) = 0;
A(9,3) = 0;
A(9,4) = -1/(Ka*L_e);
A(9,5) = 0;
A(9,6) = 0;
A(9,7) = 0;
A(9,8) = 0;
A(9,9) = -R_e/L_e;

%Input Matrix
B = zeros(9,1);
B(1,1) = 0;
B(2,1) = 0;
B(3,1) = 0;
B(4,1) = 0;
B(5,1) = 0;
B(6,1) = 0;
B(7,1) = 0;
B(8,1) = 0;
B(9,1) = 1/L_e;

%Disturbance Matrix
\[ E = \text{zeros}(9,2); \]
\[ E(1,1)= 0; \]
\[ E(1,2)= 0; \]
\[ E(2,1)= 0; \]
\[ E(2,2)= 0; \]
\[ E(3,1)= 0; \]
\[ E(3,2)= 0; \]
\[ E(4,1)= 0; \]
\[ E(4,2)= 0; \]
\[ E(5,1)= 0; \]
\[ E(5,2)= 0; \]
\[ E(6,1)= 0; \]
\[ E(6,2)= 0; \]
\[ E(7,1)= 1/J_0; \]
\[ E(7,2)= 0; \]
\[ E(8,1)= 0; \]
\[ E(8,2)= 1/J_m; \]
\[ E(9,1)= 0; \]
\[ E(9,2)= 0; \]

\%Output Matrix
\[ C = \text{zeros}(3,9); \]
\[ C(1,1)= 1; \]
\[ C(1,2)= 0; \]
\[ C(1,3)= 0; \]
\[ C(1,4)= 0; \]
\[ C(1,5)= 0; \]
\[ C(1,6)= 0; \]
\[ C(1,7)= 0; \]
\[ C(1,8)= 0; \]
\[ C(1,9)= 0; \]
\[ C(2,1)= 0; \]
\[ C(2,2)= 0; \]
\[ C(2,3)= 0; \]
\[ C(2,4)= 0; \]
\[ C(2,5)= 1; \]
\[ C(2,6)= 0; \]
\[ C(2,7)= 0; \]
\[ C(2,8)= 0; \]
\[ C(2,9)= 0; \]
\[ C(3,1)= -1*L_OTA; \]
\%C(3,1) = 0;
\[ C(3,2)= 0*L_OTA; \]
\[ C(3,3)= -L_OTA; \]
\[ C(3,4)= 2*L_OTA; \]
\[ C(3,5)= 0; \]
\[ C(3,6)= 0; \]
\[ C(3,7)= 0; \]
\[ C(3,8)= 0; \]
\[ C(1,9)= 0; \]

\%Direct Transmittance Matrix
\[ D = \text{zeros}(3,1); \]
% Altitude Structure Matrix

% System Matrix
A_a = zeros(9,9);
A_a(1,1) = 0;
A_a(1,2) = 0;
A_a(1,3) = 0;
A_a(1,4) = 0;
A_a(1,5) = 1;
A_a(1,6) = 0;
A_a(1,7) = 0;
A_a(1,8) = 0;
A_a(1,9) = 0;
A_a(2,1) = 0;
A_a(2,2) = 0;
A_a(2,3) = 0;
A_a(2,4) = 0;
A_a(2,5) = 0;
A_a(2,6) = 1;
A_a(2,7) = 0;
A_a(2,8) = 0;
A_a(2,9) = 0;
A_a(3,1) = 0;
A_a(3,2) = 0;
A_a(3,3) = 0;
A_a(3,4) = 0;
A_a(3,5) = 0;
A_a(3,6) = 0;
A_a(3,7) = 1;
A_a(3,8) = 0;
A_a(3,9) = 0;
A_a(4,1) = 0;
A_a(4,2) = 0;
A_a(4,3) = 0;
A_a(4,4) = 0;
A_a(4,5) = 0;
A_a(4,6) = 0;
A_a(4,7) = 0;
A_a(4,8) = 1;
A_a(4,9) = 0;
A_a(5,1) = -(K_o_a+K_m_a)/J_mo_a;
A_a(5,2) = 0;
A_a(5,3) = K_o_a/J_mo_a;
A_a(5,4) = K_m_a/J_mo_a;
A_a(5,5) = -(B_mo_a+B_o_a+B_m_a)/J_mo_a;
A_a(5,6) = B_mo_a/J_mo_a;
A_a(5,7) = B_o_a/J_mo_a;
A_a(5,8) = B_m_a/J_mo_a;
A_a(5,9) = 1/(Ka_a*J_mo_a);
\[ A_a(6,1) = 0; \]
\[ A_a(6,2) = -K_f_a/J_f_a; \]
\[ A_a(6,3) = 0; \]
\[ A_a(6,4) = 0; \]
\[ A_a(6,5) = B_{mo_a}/J_f_a; \]
\[ A_a(6,6) = -(B_{mo_a}+B_{f_a})/J_f_a; \]
\[ A_a(6,7) = 0; \]
\[ A_a(6,8) = 0; \]
\[ A_a(6,9) = -1/(K_{a_a} J_f_a); \]

\[ A_a(7,1) = K_o_a/J_o_a; \]
\[ A_a(7,2) = 0; \]
\[ A_a(7,3) = -K_o_a/J_o_a; \]
\[ A_a(7,4) = 0; \]
\[ A_a(7,5) = B_{o_a}/J_o_a; \]
\[ A_a(7,6) = 0; \]
\[ A_a(7,7) = -B_{o_a}/J_o_a; \]
\[ A_a(7,8) = 0; \]
\[ A_a(7,9) = 0; \]

\[ A_a(8,1) = K_m_a/J_m_a; \]
\[ A_a(8,2) = 0; \]
\[ A_a(8,3) = 0; \]
\[ A_a(8,4) = -K_m_a/J_m_a; \]
\[ A_a(8,5) = B_{m_a}/J_m_a; \]
\[ A_a(8,6) = 0; \]
\[ A_a(8,7) = 0; \]
\[ A_a(8,8) = -B_{m_a}/J_m_a; \]
\[ A_a(8,9) = 0; \]

\[ A_a(9,1) = 0; \]
\[ A_a(9,2) = 0; \]
\[ A_a(9,3) = 0; \]
\[ A_a(9,4) = 0; \]
\[ A_a(9,5) = -1/(K_{a_a} L_{e_a}); \]
\[ A_a(9,6) = 1/(K_{a_a} L_{e_a}); \]
\[ A_a(9,7) = 0; \]
\[ A_a(9,8) = 0; \]
\[ A_a(9,9) = -R_{e_a}/L_{e_a}; \]

% Input Matrix
\[ B_a = zeros(9,1); \]
\[ B_a(1,1) = 0; \]
\[ B_a(2,1) = 0; \]
\[ B_a(3,1) = 0; \]
\[ B_a(4,1) = 0; \]
\[ B_a(5,1) = 0; \]
\[ B_a(6,1) = 0; \]
\[ B_a(7,1) = 0; \]
\[ B_a(8,1) = 0; \]
\[ B_a(9,1) = 1/L_{e_a}; \]

% Disturbance Matrix
\[ E_a = zeros(9,5); \]
\[ E_a(1,1) = 0; \]
\[ E_a(1,2) = 0; \]
E_a(1,3) = 0;
E_a(1,4) = 0;
E_a(1,5) = 0;
E_a(2,1) = 0;
E_a(2,2) = 0;
E_a(2,3) = 0;
E_a(2,4) = 0;
E_a(2,5) = 0;
E_a(3,1) = 0;
E_a(3,2) = 0;
E_a(3,3) = 0;
E_a(3,4) = 0;
E_a(3,5) = 0;
E_a(4,1) = 0;
E_a(4,2) = 0;
E_a(4,3) = 0;
E_a(4,4) = 0;
E_a(4,5) = 0;
E_a(5,1) = 0;
E_a(5,2) = 0;
E_a(5,3) = 0;
E_a(5,4) = 0;
E_a(5,5) = 0;
E_a(6,1) = 0;
E_a(6,2) = 0;
E_a(6,3) = 1/J_f_a;
E_a(6,4) = 0;
E_a(6,5) = 0;
E_a(7,1) = 1/J_o_a;
E_a(7,2) = 0;
E_a(7,3) = 0;
E_a(7,4) = 1/J_o_a;
E_a(7,5) = 0;
E_a(8,1) = 0;
E_a(8,2) = 1/J_m_a;
E_a(8,3) = 0;
E_a(8,4) = 0;
E_a(8,5) = 1/J_m_a;
E_a(9,1) = 0;
E_a(9,2) = 0;
E_a(9,3) = 0;
E_a(9,4) = 0;
E_a(9,5) = 0;

%Output Matrix
C_a = zeros(3,9);
C_a(1,1) = 1;
C_a(1,2) = 0;
C_a(1,3) = 0;
C_a(1,4) = 0;
C_a(1,5) = 0;
C_a(1,6) = 0;
C_a(1,7) = 0;
C_a(1,8) = 0;
C_a(1,9) = 0;

C_a(2,1) = 0;
C_a(2,2) = 0;
C_a(2,3) = 0;
C_a(2,4) = 0;
C_a(2,5) = 1;
C_a(2,6) = 0;
C_a(2,7) = 0;
C_a(2,8) = 0;
C_a(2,9) = 0;

C_a(3,1) = -1*L_OTA;
%C_a(3,1) = 0;
C_a(3,2) = 1*L_OTA;
C_a(3,3) = -L_OTA;
C_a(3,4) = 2*L_OTA;
C_a(3,5) = 0;
C_a(3,6) = 0;
C_a(3,7) = 0;
C_a(3,8) = 0;
C_a(1,9) = 0;

%Direc Transmittance Matrix
D_a = zeros(3,1);
% Pre-run parameters
% % Chi-Yeh Hsu
% % run it before use telescope model
% %----------------------------------------------------------------------

% Plot Parameters
% %----------------------------------------------------------------------
% % Matlab Parameters
% t_end = 5;
% % % Control Input Parameters
% %----------------------------------------------------------------------
Azimuth_initial = 0;
Azimuth_final   = pi/12;
Azimuth_stepTime= 0.5;

Altitude_initial = 0;
Altitude_final   = pi/12;
Altitude_stepTime= Azimuth_stepTime;

% % Wind Parameters
% %----------------------------------------------------------------------
Wind_amplitude = 5;
Wind_period    = 10;
Wind_delay     = 6;
% % % Environmental Parameters
% %----------------------------------------------------------------------
p_air = 1.25;       % density of air (kg/m^3)
g = 0;            % Gravitational acceleration(m/s^2)
% % % Initial States
% %----------------------------------------------------------------------
% %Azimuth Initial State (11 States)
Theta_J  = 0;
Theta_Jam= Theta_J;
Theta_JO = Theta_J;
Theta_Jm = Theta_J;
Omega_J  = 0;
Omega_Jam= Omega_J;
Omega_JO = Omega_J;
Omega_Jm = Omega_J;
i_L = 0;

X_ini = [Theta_J Theta_Jam Theta_JO Theta_Jm Omega_J Omega_Jam Omega_JO Omega_Jm i_L];
%Altitude Initial State (9 States)
Theta_J_a = 0;
Theta_Jf_a = 0;
Theta_JO_a = Theta_J_a;
Theta_Jm_a = Theta_J_a;
Omega_J_a = 0;
Omega_Jf_a = Omega_J_a;
Omega_JO_a = Omega_J_a;
Omega_Jm_a = Omega_J_a;
i_L_a = 0;

X_a_ini = [Theta_J_a Theta_Jf_a Theta_JO_a Theta_Jm_a Omega_J_a
Omega_Jf_a Omega_JO_a Omega_Jm_a i_L_a];
%----------------------------------------------------------------------
% Control Parameters
%----------------------------------------------------------------------
Kp = 24;
Kd = 8;
Ki = 0;

Kp_a = 5;
Kd_a = 1.6;
Ki_a = 0;
Appendix E: Matlab Plot Functions

```matlab
% Plot function
% this function plots azimuth motion result
clc
% Plot Parameters

Y_min = min(Azimuth_System_Output(:,2))-max(Azimuth_System_Output(:,2))/10;
Y_max = max(Azimuth_System_Output(:,2))+max(Azimuth_System_Output(:,2))/10;
X_l_min = min(Azimuth_System_Output(:,4))-max(Azimuth_System_Output(:,4))/10;
X_l_max = max(Azimuth_System_Output(:,4))+max(Azimuth_System_Output(:,4))/10;
P_f = Azimuth_System_State(:,2)-Azimuth_System_State(:,1);
P_o = Azimuth_System_State(:,3)-Azimuth_System_State(:,2);
P_m = Azimuth_System_State(:,4)-Azimuth_System_State(:,2);
P_f_min = min(P_f)-max(P_f)/10;
P_f_max = max(P_f)+max(P_f)/10;
P_o_min = min(P_o)-max(P_o)/10;
P_o_max = max(P_o)+max(P_o)/10;
P_m_min = min(P_m)-max(P_m)/10;
P_m_max = max(P_m)+max(P_m)/10;
C_input = Azimuth_Control_Input(:,1);
C_input_min = min(C_input)-max(C_input)/10;
C_input_max = max(C_input)+max(C_input)/10;
Vib_max = 1*10^-5;
Vib_min = -Vib_max;

% Plot Function

figure(1);
subplot(3,2,1)
plot (Azimuth_System_Output(:,1), Azimuth_System_Output(:,2),'-');
grid on;
title(['Telescope Azimuth Rotation Position, Kp=',num2str(Kp),',
Kd=',num2str(Kd),', Ki=',num2str(Ki)]);
xlabel ('Time [sec]');
ylabel ('Position [Radius]');
axis tight
```
 subplot(3,2,3)
 plot (Azimuth_System_Output(:,1), Azimuth_System_Output(:,4),'-k');
 % Default line type is solid
 grid on;
 title ('Azimuth Image Sensor Signal Displacement');
 % You must have graph title
 xlabel ('Time [sec]');
 ylabel ('Position [M]');
 axis tight

 subplot(3,2,5)
 plot (Azimuth_System_Output(:,1), C_input, 'r');
 % Default line type is solid
 grid on;
 title ('Azimuth Control Input');
 % You must have graph title
 xlabel ('Time [sec]');
 ylabel ('Voltage (V)');
 axis tight

 subplot(3,2,2)
 plot (Azimuth_System_Output(:,1), P_f,'-');
 % Default line type is solid
 grid on;
 title ('Azimuth Fork Vibration');
 % You must have graph title
 xlabel ('Time [sec]');
 ylabel ('Displacement [radius]');
 axis tight

 subplot(3,2,4)
 plot (Azimuth_System_Output(:,1), P_o,'-');
 % Default line type is solid
 grid on;
 title ('Azimuth OTA Vibration');
 % You must have graph title
 xlabel ('Time [sec]');
 ylabel ('Displacement [radius]');
 axis tight

 subplot(3,2,6)
 plot (Azimuth_System_Output(:,1), P_m,'-');
 % Default line type is solid
 grid on;
 title ('Azimuth Mirror Vibration');
 % You must have graph title
 xlabel ('Time [sec]');
 ylabel ('Displacement [radius]');
 axis tight
% Plot function
% This function plots altitude motion result
clc
% Plot Parameters
Y_min = min(Altitude_System_Output(:,2)) - max(Altitude_System_Output(:,2))/10;
Y_max = max(Altitude_System_Output(:,2)) + max(Altitude_System_Output(:,2))/10;
X_l_min = min(Altitude_System_Output(:,4)) - max(Altitude_System_Output(:,4))/10;
X_l_max = max(Altitude_System_Output(:,4)) + max(Altitude_System_Output(:,4))/10;
P_f = Altitude_System_State(:,2);
P_o = Altitude_System_State(:,3) - Altitude_System_State(:,1);
P_m = Altitude_System_State(:,4) - Altitude_System_State(:,1);
C_input = Altitude_Control_Input(:,1);

Vib_max = 1*10^-3;
Vib_min = -Vib_max;

% Plot Function
figure(2);
subplot(3,2,1)
plot (Altitude_System_Output(:,1), Altitude_System_Output(:,2),'-');
grid on;
title ('Telescope Altitude Rotation Position, Kp=',num2str(Kp_a),', Kd=',num2str(Kd_a),', Ki=',num2str(Ki_a));
xlabel ('Time [sec]');
ylabel ('Position [Radius]');
axis tight

subplot(3,2,3)
plot (Altitude_System_Output(:,1), Altitude_System_Output(:,4),'-k');
grid on;
title ('Altitude Image Sensor Signal Displacement');
xlabel ('Time [sec]');
ylabel ('Position [M]');
axis tight

subplot(3,2,5)
plot (Altitude_System_Output(:,1), C_input, 'r');
% Default line type is solid
grid on;
title ('Control Input');
% you must have graph title
xlabel ('Time [sec]');
ylabel ('Voltage (V)');
axis tight

subplot(3,2,2)
plot (Altitude_System_Output(:,1), P_f,'-');
% Default line type is solid
grid on;
title ('Fork Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,4)
plot (Altitude_System_Output(:,1), P_o,'-');
% Default line type is solid
grid on;
title ('OTA Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight

subplot(3,2,6)
plot (Altitude_System_Output(:,1), P_m,'-');
% Default line type is solid
grid on;
title ('Mirror Vibration');
% You must have graph title
xlabel ('Time [sec]');
ylabel ('Displacement [radius]');
axis tight
Appendix F: OTA Calculation

% this function calculates Beam properties of the OTA
% Chi-Yeh Hsu
%---------------------------------------------------------------------
clear
clc

%Steel Beam
l = 450/1000;                %m
w = 38.1/1000;               %m
h = 3.175/1000;              %m
E = 200 *10^9;      %N/m^2
density = 7.85*10^3;    %kg/m^3
m = l*w*h*density    %kg

% telescope length
L_OTA = 0.45;   %m
m_OTA = 0.6;    %kg OTA
m_mm = 0.25;    %kg mirror mount
m_sen = 0.15;   %kg sensor
m_m = 0.5;      %kg mirror

I_total =
m_sen*(L_OTA*0.7)^2+m_m*(L_OTA*0.3)^2+m_OTA*(L_OTA*0.7*0.5)^2+m_mm*(L_OTA*0.3*0.5)^2  %kg*m^2
I_OTA = m_sen*(L_OTA*0.7)^2+m_OTA*(L_OTA*0.7*0.5)^2
I_mm = m_m*(L_OTA*0.3)^2+m_mm*(L_OTA*0.3*0.5)^2

%spring constant
I_o = w*(h^3)/12
K =3*E*I_o/(l)
K_1 = 2*3*E*I_o/(l*0.7)
K_2 = 2*3*E*I_o/(l*0.3)
Appendix G: System Parameters

clear
clc

%---------------------------------------------------------------------------------------
% Structure Parameters
%---------------------------------------------------------------------------------------
%Electric System-------------------------------------------------------
% Azimuth System
R_e  = 0.37;        %Internal Resistance of Motor
L_e  = 0.56/1000;   %Internal Inductance of motor
Ka = 30.58;         %Motor Constant (A/Nm or rad/V)
%Altitude System
R_e_a  = 0.37;
L_e_a  = 0.56/1000;
Ka_a = 30.58;

%Telescope Fork--------------------------------------------------------
% Azimuth System
B_f = 0.8;          %Friction (damping) of fork
K_f = 2000;         %Spring constant of fork
J_f = 0.149;        %Inertia of fork
L_Fork = 0.2;       %Length of Fork (m)
%Altitude System
B_f_a = 0.8;
K_f_a = 2000;
J_f_a = 0.149;
A_Fork_a = 0.01;

%Motor System----------------------------------------------------------
% Azimuth System
B_mo = 1.2*10^-5;   %Friction (damping) between Transmission and base
% (bearing)
J_mo = 2.1*10^-5;
J_am = J_mo+J_f;
%Altitude System
B_mo_a = 1.2*10^-5;
J_mo_a = 2.1*10^-5;

%Mirror Mount-----------------------------------------------------------
% Azimuth System
B_m = 0.2;          %Friction (damping) of mirror mount
K_m = 903;          %Spring constant of node mirror mount
J_m = 0.0103;       %Inertia of mirror mount
L_mm  = 0.05;       %Length of Mirror Mount (m)
A_mm  = 0.0075;     %Area of Mirror Mount (m^2)
%Altitude System
B_m_a = 0.2;
K_m_a = 903;
J_m_a = 0.0103;
A_mm_a = 0.0075;  % Area of mirror mount (m^2)
m_mm = 0.75;       % Mirror Mount Mass (kg)

%OTA---------------------------------------------------------------
%Azimuth System
B_o = 0.1;         % Friction (damping) of node 1(OTA)
K_o = 387;         % Spring constant of node 1(OTA)
J_o = 0.0298;      % Inertia of OTA
L_OTA = 0.45;      % Length of OTA
A_OTA = 0.06;      % Area of OTA (m^2)

%Altitude System
B_o_a = 0.1;
K_o_a = 387;
J_o_a = 0.0298;
A_OTA_a = 0.06;    % Area of OTA (m^2)
m_OTA = m_mm*L_mm/L_OTA;    % OTA mass (kg)