STATE-SPACE THERMAL CONTROL FOR CUBESATS

by

Marcus Charles Klupar

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

AEROSPACE ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2025

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Master's Committee, we certify that we have read the thesis prepared by Marcus C. Klupar, titled *State-Space Thermal Control For CubeSats* and recommend that it be accepted as fulfilling the dissertation requirement for the Master's Degree.

Ewan Douglas (Aug 14, 2025 08:49:58 PDT)
[Ewan Douglas] Date: 08/14/2025

[Daniel Larsson] Date: 08/14/2025

[Cholik Chan] Date: 08/14/2025

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to the Graduate College.

I hereby certify that I have read this thesis prepared under my direction and recommend that it be accepted as fulfilling the Master's requirement.

Ewan Douglas (Aug 14, 2025 08:49:58 PDT)
Ewan Douglas Date: 08/14/2025
Master's Thesis Committee Chair
Department of Aerospace and Mechanical Engineering

DEDICATION

To my parents, Karl and Rosemary Klupar, who have always supported me. And to UASAL, a group of people who make research fun.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Maintaining a stable thermal environment is critical for space telescopes. Effects such as thermal expansion can change the geometric relationship between optical components, impacting the accuracy of wavefront sensing and control algorithms for deformable mirrors and ultimately producing blurry images. To this end, this thesis focuses on developing a discrete state-space model for a CubeSats optical system from nonlinear governing partial differential equations. Furthermore, a feedback control methodology is used to ensure precise temperature regulation. Model results in tandem with a high-fidelity thermal simulator validate the approach taken and an in-depth discussion of results is presented.

NOMENCLATURE

$(\Delta x, \Delta y, \Delta z)$ Distance Between Nodes in Mesh

Δt Time Step

δ_{kj} Kronecker Delta

$\dot{E}_{gen}$ Energy Generation

$\dot{E}_{in}$ Energy In

$\dot{E}_{out}$ Energy Out

$\dot{E}_{st}$ Energy Storage

ϵ Emissivity

λ Eigenvalues / Poles of the System

$\dot{\mathbf{x}}(t)$ State Derivative Vector

$\mathbf{A}$ State Matrix

$\mathbf{B}$ Input Matrix

$\mathbf{C}$ Output Matrix

$\mathbf{D}$ Direct Transmission Matrix

$\mathbf{u}(t)$ Input Vector

$\mathbf{x_0}$ Initial State

$\mathbf{x}(k+1)$ State Difference Vector

$\mathbf{x}(t)$ State Vector

$\mathbf{y}(t)$ Output Vector

$\mathcal{C}$ Controllability Matrix

ρ Density

σ Stefan-Boltzmann Constant

A Surface Area

A_i Cross-Sectional Area in $i'th$ Direction

c_p Specific Heat

F_{kj} View Factor Matrix from Surface k to j

G Input Scaling Gain Matrix

$H(z)$ Transfer Function

K Gain Matrix

k Thermal Conductivity

N_i	Node i
q	Supplied Energy on Each Surface
$q_{j \to i}$	Energy from Node j to i
T_i	Temperature of Node i
V	Volume

Chapter 1

Overview & Motivation

All spacecraft launched into space require some form of temperature regulation to ensure that components do not leave their operating range. These regulation methods can be split into two forms, active and passive regulation. Active regulation requires input power to cool or heat the spacecraft, while passive regulation requires modification of natural heat dissipation [1], such as changing the optical properties of the spacecraft with paint to limit the amount of radiation absorbed and emitted [2]. An application of regulation is to correct wavefront errors in space telescopes, as these errors are magnified by temperature disturbances [3] [4] [5]. Active control is used for this task because it can return the spacecraft to its desired temperature after a slew faster than the spacecraft would passively [3] [6]. Another application is the Hubble Space Telescope; directly after launch, the telescope experienced pointing jitter partly due to the thermal expansion and contraction of the solar arrays. New arrays were installed and the control system was updated to remove excess pointing jitter [7] [8]. Active control can be open-loop without sensors or use sensors to provide informed feedback in a closed-loop environment [9]. Some methodologies of active regulation or control are PID, nonlinear, and linear state-space controllers.

PID controllers use proportional, integral, and derivative gains to minimize the error between the system output and the desired output in the frequency domain through feedback [10]. Due to their simplicity and ease of implementation, PID controllers are the most widely used [10]. However, PID controllers can become cumbersome as gain-tuning issues arise with more complex systems [11].

Real world systems, at their core, are nonlinear and thus it is appealing to model and control these systems with nonlinear control methodologies. But by simply being nonlinear, superposition is not applicable, making analysis much more difficult. In addition, controllability and observability cannot be determined by a simple rank test, and stability is no longer a function of a basic placement of eigenvalues, as done in the linear case [12]. For these reasons, nonlinear control is mostly avoided. However, nonlinear control may be the only solution if the system is highly nonlinear and no valid linear approximation will meet design requirements.

Linear state-space control is best for systems with uncertainty, instability, and exogenous inputs and has the benefits of stabilizing unstable systems, compensating for external disturbances, and correcting modeled dynamics and uncertainties. In addition, ideas such as optimal control exist as linear cost functions can be minimized [9]. Linear state-space controllers use time domain analysis to find gains that in turn modify the state or the output of the system to produce the desired response [13]. The benefit of choosing a linear state-space controller over the other options is due to its linearity. This linearity provides a separation principle that allows for control and state estimation to be designed independently [14]. Another benefit is that linear state-space control can handle multiple inputs and outputs at a time with ease [13]. Linear theory is also a well-defined field with a variety of tools to

aid in analysis. Some limitations of this method come from its linear approximation, which means that there exists a region where the approximation is not valid [13]. Another issue is pole-zero cancellation, where a mode can become 'invisible' resulting in a decoupling form the input or the output of the system [13].

The active control methodologies discussed previously can be applied to all types of spacecraft to ensure they meet their thermal requirements. However, CubeSats which are a special type of spacecraft, are growing in popularity as they enable experiments that are not possible with larger missions [15] and can be done at a cheaper cost due to 'ride-share' [16], for example DeMi [17]. A CubeSat is a small spacecraft that is used in various applications and described by its size. For example, a 1U CubeSat has a edge length of 10 cm, and larger CubeSats are described by combinations of these 1U's (e.g. 6U). While CubeSats come in various sizes, they all must follow a general design specification and meet safety standards [18][19]. In general, more CubeSats are being launched each year as an upward trend can be seen in CubeSat launches since 2005 [20]. In addition, CubeSats are becoming more advanced as smaller and more sophisticated instruments become available, and can even be used to aid other larger missions [15]. However, as with larger space telescopes, thermal stability is still needed [15] [21]. Thermal control is especially needed for CubeSats that carry optical payloads, as thermal flutter and thermal gradients can cause wavefront jitter and produce instrument de-focus [22] [23]. For some applications, even $\pm 1^{\circ}$C can cause unwanted deformation of the optics [24]. Prudent thermal regulation is even more important for CubeSats carrying deformable mirrors as any thermal gradient can cause phase errors in the wavefront. These phase errors can be corrected with adaptive optics, but the task is made easier if thermal effects are minimal [25].

To provide linear control that removes unwanted thermal effects for these CubeSats, a thermal model of the CubeSat will be needed. These thermal models are approximations of the actual CubeSats governing equations and can be created in a variety of ways. The most popular is the Finite Element Method (FEM), where an intricate domain is discretized into smaller elements, called a mesh. A solution is found for each element and then combined to arrive at the full solution [26] [27]. The first application of this method was to solve problems related to stiffness and deflection of airplane wings, and was later used in stress analysis and heat transfer [27] [28] [29] [30]. The main appeal of this method is that the boundary and interior can be adequately approximated by triangular elements, and as these elements become smaller, the approximate solution converges to the analytical solution [26][31]. The biggest difficulty with this method is how the meshes are created, as this is the most computationally demanding part [32].

Another less popular method that achieves the same goal as FEM is called the Boundary Element Method (BEM). Instead of discretizing the entire domain, only the boundary is discretized. This allows for a reduction in spatial dimension, leading to simpler solutions and less computational cost as no interior mesh is needed [33]. Two benefits of this method are that only the boundary needs to be meshed and that a continuous solution exists, unlike in FEM. These benefits come from the properties of Green's functions and the divergence theorem [33] [34]. This method is sometimes avoided due to the advanced mathematical knowledge required to formulate a solution.

The last method is the Finite Difference Method (FDM), which again is analogous to FEM and BEM. In this method, derivatives are approximated over small temporal and spatial regions with a square mesh[35]. The main difference between FDM and FEM is that

FEM allows for other non-square mesh types, which leads to better boundary modeling [26]. FDM has also been used to solve partial differential equations, such as the heat equation [36] [37] [35]. The main benefit and fault of this method is its simplicity to implement and understand, however, for larger systems computational difficulties arise [38].

In this thesis we will use the finite difference method to create a thermal model of the CubeSat, and then use linear control theory to regulate the temperature of the optics and remove unwanted thermal effects.

Chapter 2

Linear Systems and State Space Control

2.1 Discrete Linear Systems

A continuous linear state-space system is of the form,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{2.1}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t),$$

where $\mathbf{x}(t) \in \mathbb{R}^{n\times 1}$ is the state of the system, $\mathbf{u}(t) \in \mathbb{R}^{m\times 1}$ is the system input vector, and $\mathbf{y}(t) \in \mathbb{R}^{l\times 1}$ is the system output vector.

The matrix $A \in \mathbb{R}^{n\times n}$ is called the state matrix, and it embodies the entire dynamics of the system. $B \in \mathbb{R}^{n\times m}$ is a matrix that represents how inputs enter the system and is called the input matrix. $C \in \mathbb{R}^{l\times n}$ is a matrix that determines how states appear in the output and is called the output matrix. Finally, $D \in \mathbb{R}^{l\times m}$ is a matrix that allows an input to bypass the system and go directly to the output and is sometimes called the direct transmission matrix [13, p. 239].

To derive a system representation in the form of equation 2.1, a n^{th} order ordinary differential equation that describes the system is needed. This n^{th} order ODE can be written as

a system of first-order ODEs through the introduction of state variables. Take, for example, a simple 2nd order ODE,

$$\ddot{v}(t) = a\dot{v}(t) + bv(t) + cu(t), \tag{2.2}$$

we can next introduce the state variables $x_1(t)$ and $x_2(t)$,

$$x_1(t) = v(t), \tag{2.3}$$

$$\dot{x}_1(t) = \dot{v}(t) = x_2(t), \tag{2.4}$$

$$\dot{x}_2(t) = \ddot{v}(t). \tag{2.5}$$

Finally, substituting these state variables into our ODE results in,

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ b & a \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ c \end{bmatrix} u(t), \tag{2.6}$$
$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

Through the use of state variables, we are able to represent the ODE in the same form as equation 2.1.

However, continuous systems are not often encountered in practice, as the system's inputs and outputs are measured at discrete times rather than continuously. It can be shown that by sampling the continuous linear system we can arrive at a time-invariant discrete-time linear system [13, p. 265]. Discrete linear systems are of the form,

$$\mathbf{x}_{k+1} = A_d\mathbf{x}_k + B_d\mathbf{u}_k, \tag{2.7}$$

$$\mathbf{y}_k = C_d\mathbf{x}_k + D_d\mathbf{u}_k.$$

This notation is similar to that of the continuous system except that the system is evaluated at discrete values of k, which represent time. The size and meaning of each vector and matrix is the same as in the continuous-time system. One difference between continuous and discrete times systems is the mathematical transform used to simplify the equations. In continuous time, the Laplace transform is used to transform the system from the time domain to the frequency domain. In discrete time, the z-transform is used to accomplish the same goal where 'z' is a time delay operator. The subscript d is used to emphasize that equation 2.7 is discrete. The subscript d on the matrices will be dropped as we will only be dealing with discrete linear systems. As seen in the next section, the properties of the A,B, and C matrices illustrate how the system will behave. The linear system described by A, B, and C can equivalently be represented by a block diagram, which can help visualize the math [39]. The D matrix is not included since it can be viewed as an output shift. This output shift allows us to perform our usual controls analysis on just the A,B, and C matrices. Once the analysis is complete we can undo our original shift to get the correct output.

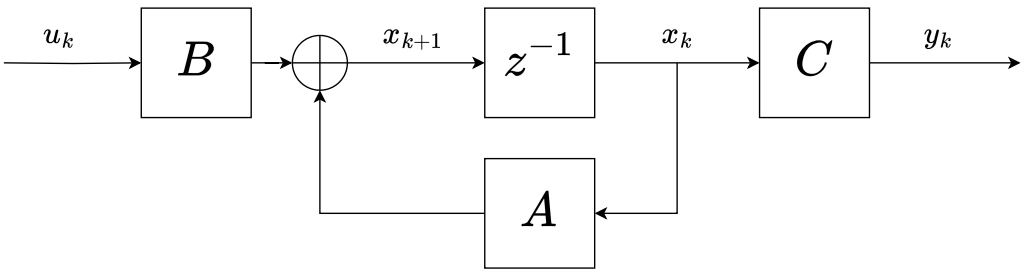

Figure 2.1: Block Diagram illustrating how the input $\mathbf{u}_k$ reaches the output $\mathbf{y}_k$ of the discrete time system from equation 2.7. The z-transform, z^{-1}, acts as a time delay operator.

2.2 Linear System Properties

2.2.1 Asymptotic Stability

Stability is a property that focuses on the the equilibrium points of the state-space system. This property identifies the tendency for a state to approach its equilibrium point or to diverge to infinity. For a state-space system to be asymptotically stable, we say that the state's asymptotically approach the equilibrium point. To quantify the behavior around these equilibrium points, we need an equation for the state. To do this we take equation 2.7, and write it out for the first few time-steps. Through induction one can show that,

$$\mathbf{x}_k = A^k \mathbf{x}_0 + \sum_{i=0}^{k-1} A^{k-i-1} B \mathbf{u}_i. \tag{2.8}$$

We can observe the tendency of the state, or its free response, by taking $\mathbf{u}_i = 0$. This leaves $\mathbf{x}_k = A^k \mathbf{x}_0$, which will decay to zero if the eigenvalues of A have a magnitude less than 1. The decay to zero can be observed by noting that any number with magnitude less than one will decay towards zero if it is recursively multiplied to itself. If we decompose A into its Jordan form, we will see that the diagonal elements of the Jordan form of A, the eigenvalues,

are being recursively multiplied when taking A^k. As k tends to infinity A^k will tend to zero, resulting in $\lim_{x\to\infty} ||\mathbf{x}_k|| = \mathbf{0}$. If this condition is true, then the system defined by A is said to be asymptotically stable [13, p. 296]. In addition, we say that if the eigenvalues are inside the unit circle, then they are stable eigenvalues, and if they are outside the unit circle then they are unstable eigenvalues.

Determining the stability of any given system in the form of equation 2.7 will help determine how the system behaves. In addition, stability analysis can help us determine the unstable states that cause problems when it comes to implementing a control law. The topics of controllability, stabilizability, and control laws are discussed in the sections to come and will help facilitate the importance of stability.

2.2.2 Controllability

To determine to what extent we can influence the state through a input sequence u_i, we next examine controllability. The idea put simply is: can the initial state vector $\mathbf{x}_0$ be moved to any other position $\mathbf{x}_{final}$ in a finite amount of time by means of a control sequence $\mathbf{u}_k$. Intuitively, a mathematical expression for controllability should depend on the state and input matrices, as $\mathbf{u}_k$, the input vector, is used to move the state through a combination of A and B in equation 2.9 [13, p. 302].

To determine if the system from equation 2.7 is controllable, a controllability matrix is formed,

$$\mathcal{C} = [B, AB, ..., A^{n-1}B], \tag{2.9}$$

which comes from writing $\sum_{i=0}^{k-1} A^{k-i-1}B\mathbf{u}_i$ from equation 2.8 as the dot product $\mathcal{C}\cdot[\mathbf{u}_{k-1},\ldots,\mathbf{u}_0]^T$ [13, p. 304].. If $\text{rank}(\mathcal{C}) = n$, then the system is controllable. This makes sense as if $\mathcal{C}$ is full rank then each input will effect each state of the system. If $\text{rank}(\mathcal{C}) = p < n$, then there are $n - p$ uncontrollable states, which cannot be influenced through a control input. A formal proof for this is found in [13, p. 304].

It is important to know if the system from equation 2.7 is controllable because it tells us if we can change the state through the inputs $\mathbf{u}_k$. For example, if a state is unstable and also cannot be controlled to a final position through $\mathbf{u}_k$, then there is nothing we can do to regulate this state with the current system.

2.2.3 Stabilizability and Uncontrollable Form

The last property discussed is stabilizability. This idea adds a extra layer of analysis for when a state is not controllable. If a state is not controllable, i.e. we cannot move the the state to a desired final value through the input, then we hope that the eigenvalue of that uncontrollable state is already stable. This ensures that the uncontrollable states do not diverge to infinity. To identify the uncontrollable states, one can use a transformation matrix to transform the system into its uncontrollable form. We can use this transformation only if $\text{rank}(\mathcal{C}) \neq n$, i.e. the system is already identified to be uncontrollable. The pair (A, B) can be written in the uncontrollable form as,

$$A = \begin{bmatrix} A_c & A_{cuc} \\ 0 & A_{uc} \end{bmatrix} \quad B = \begin{bmatrix} B_c \\ 0 \end{bmatrix}, \tag{2.10}$$

Where the pair (A_c, B_c) is controllable and thus its eigenvalues can be made stable and A_{uc} is the uncontrollable portion of A. Therefore, our system is stabalizable if the eigenvalues of A_{uc} are within the unit circle. A proof for this form is found in [13, p. 310]. If A_{uc} has eigenvalues outside the unit circle, then the physical system needs to be modified to remove these non-stabalizable states.

2.3 Control Laws and Feedback

The idea of a control sequence $\mathbf{u}_k$ to control the state from $\mathbf{x}_0$ (initial state) to $\mathbf{x}_{final}$ (final state) was previously discussed in the controllability subsection. To determine the correct input to move $\mathbf{x}_0$, we use the general form of the linear state feedback control law. This control law is,

$$\mathbf{u_k} = G\mathbf{r_k} - K\mathbf{x_k}, \tag{2.11}$$

where $K \in \mathbb{R}^{m\times n}$ and $G \in \mathbb{R}^{m\times n}$ are gain matrices and $\mathbf{r}_k \in \mathbb{R}^{n\times 1}$ is a reference signal. This control law is one where we design K to give a desired transient response, and design G to provide steady-state response.

If the system is controllable, then there exists a $\mathbf{u}_k$ that will allow the initial state to be moved to any final state in finite time [13, p. 302]. If $\mathbf{u_k}$ is substituted into equation 2.7 and assuming $D = 0$, we arrive at a close-loop form for the system,

$$\mathbf{x}_{k+1} = (A - BK)\mathbf{x}_k + (BG)\mathbf{r}_k \tag{2.12}$$

$$\mathbf{y}_k = C\mathbf{x}_k$$

From this we see that the closed-loop eigenvalues are those of the system $A_{cl} = (A - BK)$. One way to design K is through pole placement. This allows the system designer to dictate how the system responds, and it involves matching the closed-loop characteristic equation to that of the desired characteristic equation [13, p. 351]. This is discussed more in the next section. In short, the control law 2.11 allows the state to alter the dynamics of the system via the gain matrix K. Given the system is controllable, we can arbitrarily change the closed-loop eigenvalues of A_{cl} to give a desired response through the proper choice of K. This is a significant result as A could have eigenvalues outside the unit circle and therefore have a non-stable open-loop response. However, if the pair (A, B) is controllable, then the eigenvalues of A_{cl} can be placed inside the unit circle through the pole placement method, producing a stable closed-loop response. If (A, B) is not controllable, then the current system can not be controlled to produce a desired response through the input $\mathbf{u}_k$. This control law can be visualized in Figure 2.1 by picking off the state and multiplying it by a K block, then taking the difference with $G\mathbf{r}_k$ [39]. To determine the values of K and G, we explore the pole placement method and the servo problem in the next two sections.

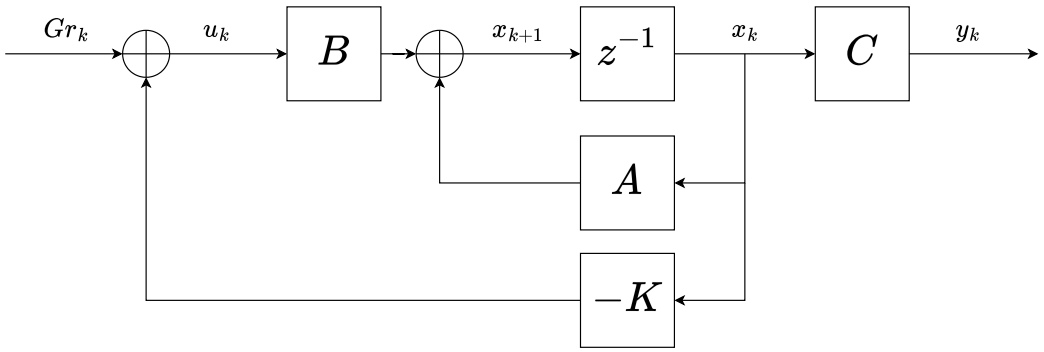

Figure 2.2: Discrete Time Block Diagram representing equation 2.12. The block $-K$ is used for state feedback and G is used for steady state tracking.

2.4 Pole Placement

It was seen in the previous section that the choice of K directly affects the closed-loop dynamics. One method of choosing the matrix K is called pole placement and is performed by selecting a gain matrix to produce the desired eigenvalues of A_{cl}.

If the system has a single input and is controllable, then there exists a gain matrix that arbitrarily places the system poles [13, p. 353]. To determine K for a single input system,

1. Identify the closed-loop characteristic polynomial.

   $det[\lambda I_n - (A - BK)]$

2. Determine the characteristic polynomial for the desired eigenvalues, where λ_i is the desired eigenvalue.

   $\Pi_{i=1}^{n}(\lambda - \lambda_i)$

3. Set the two characteristic polynomial's equal to each other and solve for K. [13, p. 355].

If the system has multiple inputs and is controllable, then we can determine the gain matrix through Theorem 1 [13, p. 365].

Theorem 1 *For any controllable pair (A, B), there exists a state feedback matrix K that assigns the eigen-pairs $\{\lambda_i, \mathbf{v}_{di}\}, i = 1, \ldots, n$ if and only if*

$$U_1^T[A - \lambda_i I_n]\mathbf{v}_{di} = \mathbf{0}_{n\times 1} \tag{2.13}$$

where

$$B = U \begin{bmatrix} \Sigma_B V^T \\ \\ \mathbf{0}_{n\times 1} \end{bmatrix} \tag{2.14}$$

with

$$U = \begin{bmatrix} U_0 & | & U_1 \end{bmatrix}, U^{-1} = U^T. \tag{2.15}$$

The state feedback matrix is then given by,

$$K = -V\Sigma_B^{-1}U_0^T[V_d\Lambda V_d^{-1} - A], \tag{2.16}$$

where $V_d\Lambda V_d^{-1} = A_{cl}$, U and V are the singular matrices for conducting a singular value decomposition on B, and Σ_B is the non-zero elements of B.

The method of pole placement is a powerful tool that allows us to design the closed-loop response. Through the choice of K, we can arbitrarily place the eigenvalues of our system to produce a desired closed-loop response.

2.5 Servo Problem: Steady-State Tracking

Now that we have determined how to calculate K to change the transient response, we need to calculate G to produce a desired steady-state response. This is often referred to as the servo problem in literature [13, p. 367]. The desired steady-state response that we are seeking to create is one where the state tracks the reference signal with no steady-state error. To derive G, we start with the control law,

$$\mathbf{u}_k = G\mathbf{r}_k - K\mathbf{x}_k, \tag{2.17}$$

and form the closed-loop system by substituting the control law into equation 2.7,

$$\begin{aligned} \mathbf{x}_{k+1} &= (A - BK)\mathbf{x}_k + (BG)\mathbf{r}_k, \\ \mathbf{y}_k &= C\mathbf{x}_k. \end{aligned} \tag{2.18}$$

To determine the value of G for steady-state tracking, we take the z-transform of 2.18 with $x_0 = 0$,

$$z\mathbf{X}(z) = (A - BK)\mathbf{X}(z) + BG\mathbf{R}(z), \tag{2.19}$$

$$\mathbf{Y}(z) = C\mathbf{X}(z), \tag{2.20}$$

and then isolate for $\mathbf{X}(z)$ and plug the result into the equation for $\mathbf{Y}(z)$,

$$\mathbf{X}(z) = [zI - (A - BK)]^{-1}BG\mathbf{R}(z), \tag{2.21}$$

$$\mathbf{Y}(z) = C\mathbf{X}(z) = C[zI - (A - BK)]^{-1}BG\mathbf{R}(z). \tag{2.22}$$

Note that the discrete transfer function $H(z)$ relates the output to the input by,

$$\mathbf{Y}(z) = H(z)\mathbf{R}(z). \tag{2.23}$$

Comparing the two previous equations results in the closed-loop transfer function,

$$H(z) = C[zI - (A - BK)]^{-1}BG. \tag{2.24}$$

Since we want the steady-state response to be constant we set $\mathbf{R}(z)$ to be a step input of magnitude M,

$$\mathbf{R}(z) = M\frac{z}{z-1}, \tag{2.25}$$

$$\mathbf{Y}(z) = C[zI - (A - BK)]^{-1}BGM\frac{z}{z-1}. \tag{2.26}$$

Additionally, the final value theorem for discrete time systems, which determines the final output of the system when time equals infinity, is used,

$$\mathbf{Y}_{ss} = \lim_{z\to 1}(z-1)C[zI-(A-BK)]^{-1}BGM\frac{z}{z-1}, \tag{2.27}$$

$$= \lim_{z\to 1} C[zI-(A-BK)]^{-1}BGMz, \tag{2.28}$$

$$= C[I-(A-BK)]^{-1}BGM. \tag{2.29}$$

For zero steady-state error, we want the final value of our system to be M ($\mathbf{Y}_{ss} = M$), which results in,

$$I = C[I-(A-BK)]^{-1}BG, \tag{2.30}$$

and finally solving for G gives,

$$G = (C[I-(A-BK)]^{-1}B)^{-1}. \tag{2.31}$$

It is important to observe that $C[I-(A-BK)]^{-1}B \in \mathbb{R}^{n\times n}$ ($l = m = n$) for G to track all the reference signals with no steady-state error. Or, in the case where $l \neq m$, i.e. $C[I-(A-BK)]^{-1}B$ is not square, we can still track all the reference signals if $C[I-(A-BK)]^{-1}B$ has a right inverse,

$$(C[I-(A-BK)]^{-1}B)_R = G. \tag{2.32}$$

If a right inverse does not exist, a minimum-norm G can be found through the least squares method. Starting with

$$I = C[I - (A - BK)]^{-1}BG \tag{2.33}$$

and making the following variable substitution,

$$V = C[I - (A - BK)]^{-1}B \tag{2.34}$$

leads to,

$$\min_{G} ||I - VG||^2. \tag{2.35}$$

If a minimum norm G is found to make $||I - VG||^2 = 0$, then perfect steady-state tracking can be achieved. However, if the minimum norm G does not provide $||I - VG||^2 = 0$, then only approximate steady-state tracking can be achieved.

2.6 Linearization of Nonlinear State-Space Equations

The real world is inherently nonlinear which means that the equations used to describe real world dynamics are also nonlinear. This nonlinearity prevents us from directly using the methods previously described in this chapter. However, if we can approximate the nonlinear dynamics as linear, then we can used our tool box of methods.

To linearize a nonlinear state-space equation we first need to define equilibrium points. An equilibrium point is one where the state will not deviate from unless it is externally perturbed

[13, p. 244]. In Section 2.2.1 we defined a asymptotically stable system as one where "the state asymptotically approaches the equilibrium point." If the state of the system starts at this equilibrium point then it will not deviate from it unless it is perturbed.

Next, we define our nonlinear state-space equation as,

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{2.36}$$

where $f(\mathbf{x}_k, \mathbf{u}_k)$ is a nonlinear function of the state $\mathbf{x}_k$ and the input $\mathbf{u}_k$. To linearize this equation we take a first order Taylor series of $f(\mathbf{x}_k, \mathbf{u}_k)$ about the equilibrium point $(\mathbf{x}_{eq}, \mathbf{u}_{eq})$,

$$f(\mathbf{x}_k, \mathbf{u}_k) \approx f(\mathbf{x}_{eq}, \mathbf{u}_{eq}) + \frac{\partial f}{\partial \mathbf{x}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})} \Delta\mathbf{x}_k + \frac{\partial f}{\partial \mathbf{u}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})} \Delta\mathbf{u}_k. \tag{2.37}$$

Moving $f(\mathbf{x}_{eq}, \mathbf{u}_{eq})$ to the left hand side results in,

$$\Delta\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) - f(\mathbf{x}_{eq}, \mathbf{u}_{eq}) = \frac{\partial f}{\partial \mathbf{x}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})} \Delta\mathbf{x}_k + \frac{\partial f}{\partial \mathbf{u}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})} \Delta\mathbf{u}_k, \tag{2.38}$$

where $\Delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_{eq}$ and $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{eq}$.

Finally, if we define $A = \frac{\partial f}{\partial \mathbf{x}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})}$ and $B = \frac{\partial f}{\partial \mathbf{u}_k}|_{(\mathbf{x}_{eq}, \mathbf{u}_{eq})}$ then we arrive at

$$\Delta\mathbf{x}_{k+1} = A\Delta\mathbf{x}_k + B\Delta\mathbf{u}_k. \tag{2.39}$$

Equation 2.39 is functionally the same as equation 2.7 except for that equation 2.39 is a perturbed system defined around the equilibrium point (i.e we shift $\mathbf{x}_0$ to $\mathbf{x}_{eq}$)[13, p.243]. It should also be noted that equation 2.39 is only valid in regions where the linearization is a

"good enough" approximation of the nonlinear dynamics. "Good enough" is a problem of its own and is often quantified by engineering constraints.

Chapter 3

Finite Difference Thermal Modeling

3.1 Continuous to Discrete

As mentioned in the Overview & Motivation Chapter, we will need a thermal model of the CubeSat to carry out linear control. To do this, we begin with the governing methods for heat transfer, which always start with a control volume and its energy balance. A general energy balance, expressed in terms of energy rates, is given by [40, p. 14],

$$\dot{E}_{st} = \dot{E}_{in} - \dot{E}_{out} + \dot{E}_{gen}, \tag{3.1}$$

where $\dot{E}_{st}$ is a energy storage term, $\dot{E}_{in}$ and $\dot{E}_{out}$ are the energy entering and leaving the control volume, and $\dot{E}_{gen}$ is the internal heat generation of the control volume. Each of these terms, as we will see soon, is described by a partial derivative with respect to time ($\frac{\partial T}{\partial t}$) or space ($\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}$).

Our ultimate objective is to express equation 3.1 in state-space form, but doing so would require an infinite number of states due to the presence of partial derivatives. To avoid the need for infinite states, the partial derivatives can be approximated and evaluated at discrete points. The formulation of the discretize heat equation is the topic of the next section.

3.2 Energy Balance Method

In order to use a finite difference approximation for the derivatives, one must define where to evaluate those derivatives. To do this, a mesh is created by 'slicing' the system into smaller sections. This 'slicing' action is done to discretize the domain into smaller elements called control volumes which allow us to approximate the spatial derivatives. We will also need to discretize time, which will allows us to also approximate the time derivatives. Each control volume has its own spatial and time-dependent properties which are encompassed by a node at its center.

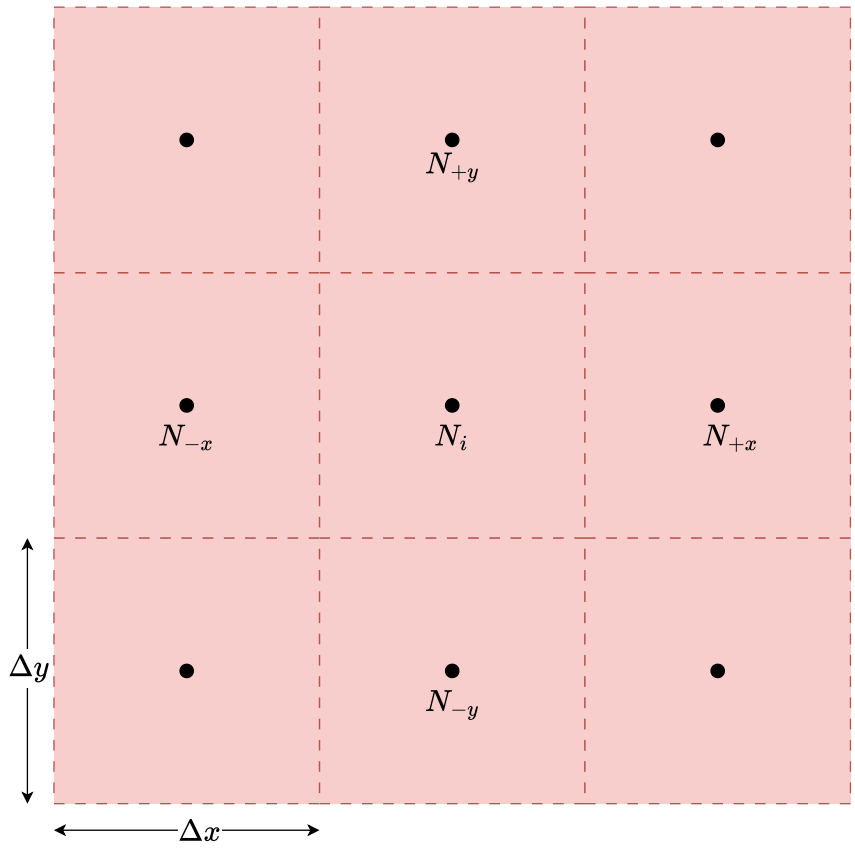

Figure 3.1: Generic mesh of the systems domain. The mesh is comprised of control volumes (dashed red lines), and nodes (black dots) that represent the systems physical properties $(T, k, \rho, V, ...)$. Square control volumes occurs when $\Delta x = \Delta y$.

The finite difference approximations for the partial derivative are evaluated with a central-difference. Therefore, the approximations for $\frac{\partial T}{\partial x}$, $\frac{\partial T}{\partial y}$, and $\frac{\partial T}{\partial z}$ are evaluated at the shared boundary between two adjacent nodes [40, p. 242]. A similar expression is true for $\frac{\partial T}{\partial t}$, except that a forward difference is used. To keep the derivations straight forward, we only outline the procedure for a two-dimensional mesh. However, we will expand the final result to three dimensions at the end.

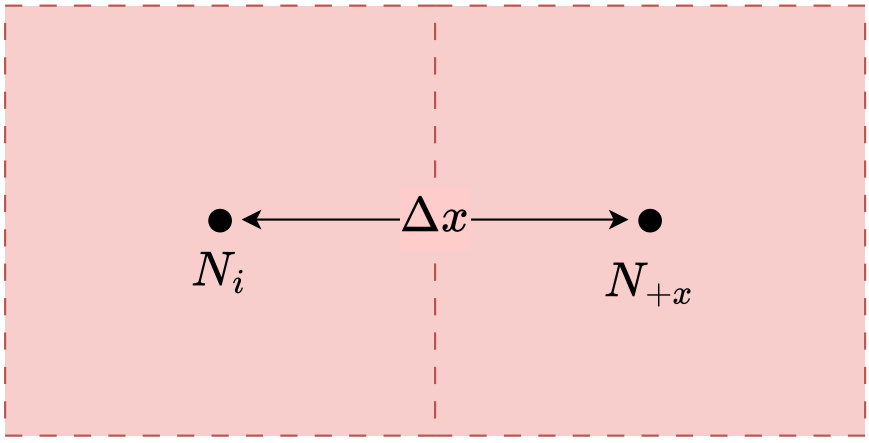

Figure 3.2: Depiction of nodes (N_i) and (N_{+x}) and the distance between them (Δx) and used to help define the central-difference approximation for $\frac{\partial T}{\partial x}$.

Using the central-difference approximation for nodes N_i and N_{+x} leads to,

$$\frac{\partial T}{\partial x} = \frac{T^p_{+x} - T^p_i}{\Delta x}. \tag{3.2}$$

Therefore, the energy due to conduction from node N_i to N_{+x} is,

$$q_{i\to(+x)} = -kA_x \frac{T^p_{+x} - T^p_i}{\Delta x}, \tag{3.3}$$

where A_x is the cross-sectional area between nodes N_i ans N_{+x}. Notice that $q_{i\to+x}$ is positive when energy is moving in the positive x-direction.

A similar equation can also be written for time,

$$\frac{\partial T}{\partial t} = \frac{T_i^{p+1} - T_i^p}{\Delta t}, \tag{3.4}$$

where the superscript p is used to represent the current time step.

With approximations for all the partial derivatives, we can now write equation 3.1 for node N_i as a finite difference. Since the direction of heat transfer is not known, it is common to assume that the heat flow is into the control volume of node N_i. Doing so causes the energy balance to become [40, p. 243],

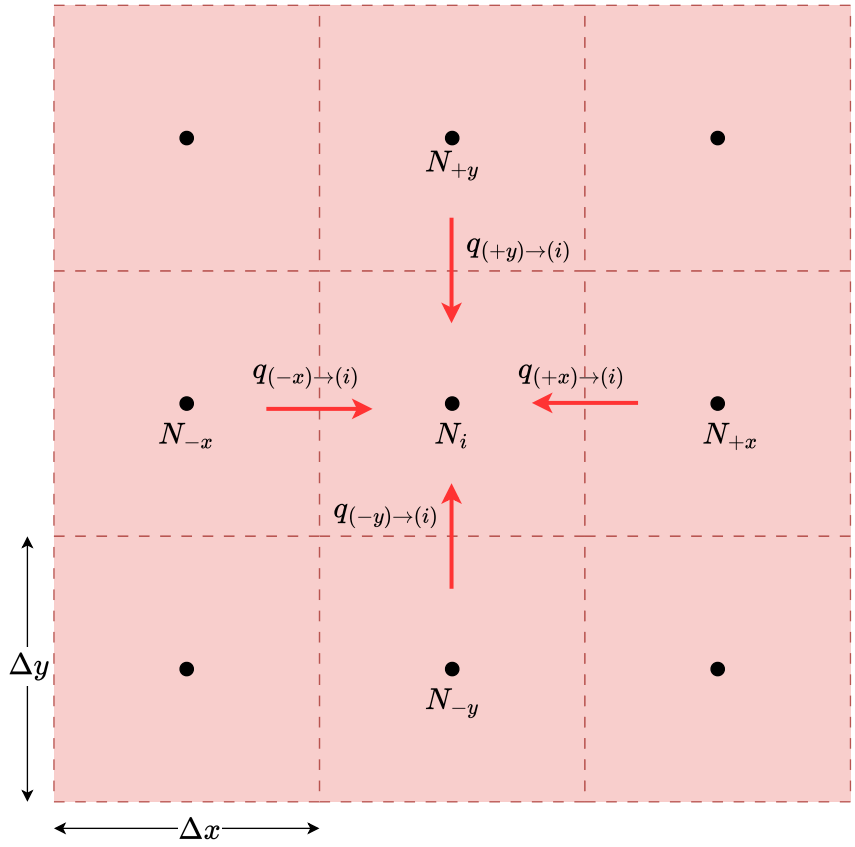

Figure 3.3: Complete energy balance for node N_i, showing the energy entering its control volume by conduction from its surrounding nodes.

$$\dot{E}_{in} + \dot{E}_{gen} = \dot{E}_{st}$$

$$\sum_{j=1}^{4} q_{j\to i} + \dot{E}_{gen} = \rho c_p V \frac{\partial T_i}{\partial t}$$

Here, the sum represents all the energy entering the control volume of node N_i from the N_j surrounding nodes. Since we only consider conduction at this moment, $q_{j\to i} = -kA_r \frac{T_i^p - T_j^p}{\Delta r}$, where r is the spatial direction between the nodes N_j and N_i, e.g (r = x,y, or z). Expanding

the energy balance above to include the z-direction leads to the following finite difference equation with $\gamma = \frac{1}{\rho c_p V}$,

$$\begin{aligned} T_i^{p+1} = \gamma k A_x \Delta t \left[\frac{T_{-x}^p + T_{+x}^p - 2T_i^p}{\Delta x}\right] \\ + \gamma k A_y \Delta t \left[\frac{T_{-y}^p + T_{+y}^p - 2T_i^p}{\Delta y}\right] \\ + \gamma k A_z \Delta t \left[\frac{T_{-z}^p + T_{+z}^p - 2T_i^p}{\Delta z}\right] \\ + \gamma \Delta t \dot{E}_{gen}^p + T_i^p. \end{aligned} \tag{3.5}$$

Equation 3.5 can then be written for all nodes in the mesh, creating a system of equations. The system of equations states that for any given node N_i, its future temperature depends on the nodes surrounding it and itself. The temperature profile for each node can be found by marching through time and solving the system of equations at each time step. It should be noted that the solution only converges if the coefficient paired with the current node of interest at the previous time is greater than or equal to zero [40, p. 331]. In equation 3.5, the current node of interest at the previous time is T_i^p. We can find the coefficient of this node by collecting like terms and then factoring out T_i^p. Therefore, the coefficient for T_i^p is,

$$1 - 2\gamma k \Delta t \left(\frac{A_x}{\Delta x} + \frac{A_y}{\Delta y} + \frac{A_z}{\Delta z}\right). \tag{3.6}$$

Next, enforcing that this coefficient is greater than or equal to zero leads to,

$$\frac{1}{2\gamma k(\frac{A_x}{\Delta x}+\frac{A_y}{\Delta y}+\frac{A_z}{\Delta z})} \geq \Delta t. \tag{3.7}$$

Hence, by choosing a discretizations for our mesh $(\Delta x, \Delta y, \Delta z)$, we have also chosen the max allowable timestep that will have a stable solution.

Chapter 4

Radiative Thermal Modeling

4.1 Net Radiation Method

A mode of heat transfer that we did not include in our finite difference analysis is radiation. Since our CubeSat will be placed in space, it is important to provide an analysis for this mode of heat transfer. To identify the radiation exchange between each surface of a generic system, a diffuse, gray, and uniform radiosity enclosure is assumed. This enclosure is sectioned into surfaces that are assumed to be isothermal. Each surface has surface properties such as emissivity ϵ_j, area A_j, temperature T_j, and a supplied energy q_j that maintains the surface j at its temperature. Or q_j can be viewed as the net radiation loss from surface j[41, p. 218 - 220].

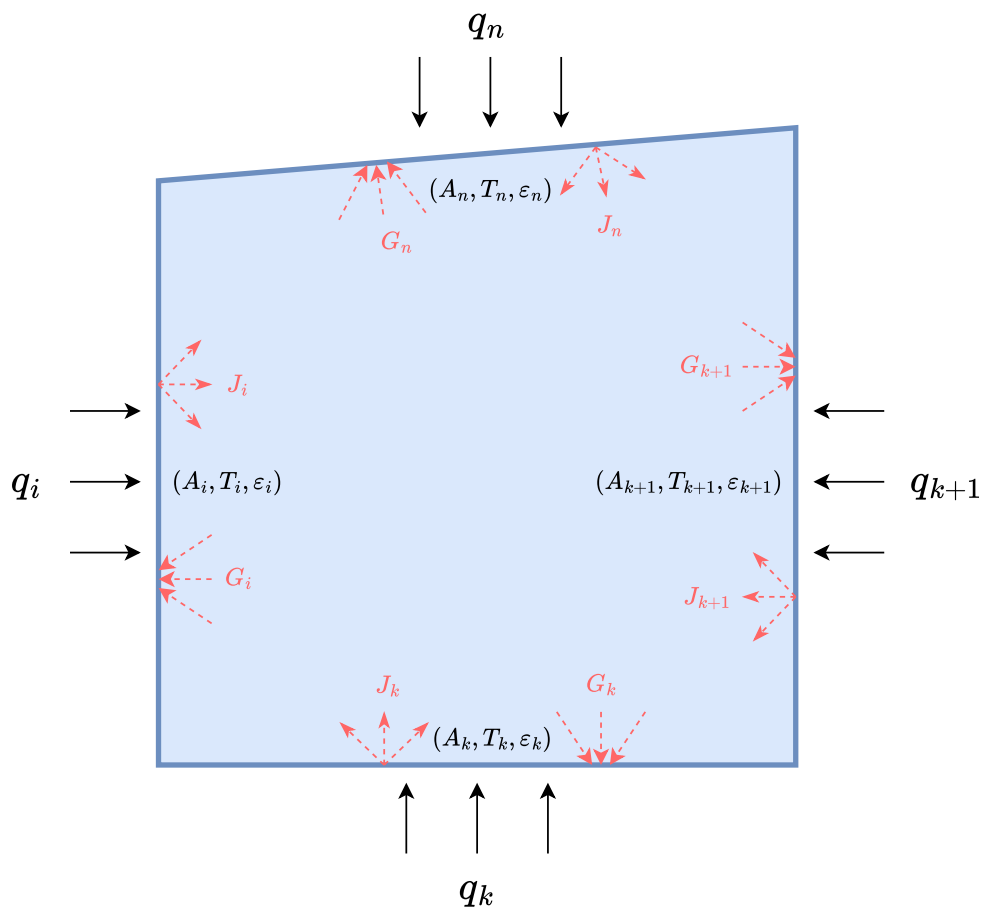

Figure 4.1: Generic enclosure depicting how the isothermal surfaces interact through incoming radiation from all surfaces (G) and outgoing radiation from a given surface (J) . A consequence of assuming that each surface is diffuse, gray, and emits uniformly is that $\epsilon = \gamma$. q_j represented all the energy supplied to surface j through means other than internal radiation. For transient solutions, q_j also represents the net ration loss from surface j.

From the above figure it is clear that each surface in the enclosure will emit, absorb, and reflect radiation. This complex behavior can be modeled by observing the net radiation entering and leaving each surface as done by [41, p. 224]. Doing so results in the following

system of equations,

$$\sum_{j=1}^{n}\left(\frac{\delta_{kj}}{\epsilon_j}-F_{kj}\frac{1-\epsilon_j}{\epsilon_j}\right)\frac{q_j}{A_j}=\sum_{j=1}^{n}(\delta_{kj}-F_{kj})\sigma T_j^4 \tag{4.1}$$

The index n is the number of surfaces in the enclosure and also the number of equations. The subscripts kj define the surfaces that interact, so F_{kj} is the view factor from surface k to surface j. It is important to note that in order for this system of equations to be solved, each surface must have either a temperature (T_j) or rate of energy flow (q_j) boundary condition defined for it.

Equation 4.1 is not only for steady-state calculations. It can be used for transient solutions since q_j is the instantaneous net radiation loss for each surface, meaning that it can match with other boundary conditions [40, p. 212]. This lets q_j naturally fit into any transient finite difference equation.

4.2 Finite Difference Equation with Radiation

Now that the radiation between surfaces can be modeled, equation 3.1 needs to be altered to include radiation. The only alteration needed is,

$$\dot{E}_{in}=\sum_{j=1}^{4}q_{j\to i}-q_{i,rad}, \tag{4.2}$$

where $q_{i,rad}$ is the instantaneous net radiation loss from the external surface associated with node N_i. The minus sign is due to the radiation, $q_{i,rad}$, being defined in the outward direction of the control volume.

Equation 3.5 now becomes,

$$
\begin{aligned}
T_i^{(p+1)} = {} & \gamma k A_x \Delta t \left[\frac{T_{-x}^p + T_{+x}^p - 2T_i^p}{\Delta x} \right] \\
& + \gamma k A_y \Delta t \left[\frac{T_{-y}^p + T_{+y}^p - 2T_i^p}{\Delta y} \right] \\
& + \gamma k A_z \Delta t \left[\frac{T_{-z}^p + T_{+z}^p - 2T_i^p}{\Delta z} \right] \\
& + \gamma \Delta t \dot{E}_{gen}^p - \gamma \Delta t q_{i,rad}^p + T_i^p
\end{aligned} \tag{4.3}
$$

where $q_{i,rad}$ is solved for using equation 4.1. Equation 4.3 can be solved for each node by stepping through time. Since the radiation term is nonlinear, it will need to be linearized in order to create a linear state-space representation. This can be done easily by using a second-order Taylor series.

Chapter 5

Thermal Desktop Modeling: 1 Node Case

5.1 Model Creation

Thermal Desktop is used as a means to validate the thermal modeling done in this thesis. Thermal Desktop is a thermal finite element and finite difference software that uses SINDA/FLUENT as its solver. It is exceptional at modeling all modes of heat transfer and simplifying design processes. The process for creating a CubeSat and its optical system is illustrated in the following sections, and it is used as a means to validate the finite difference calculations done in MATLAB.

5.1.1 Defining Properties

Once Thermal Desktop is open, the first thing to do is define the thermal and optical properties. This can be done by selecting 'Thermal > Optical Properties > Edit Property Data' in the top ribbon bar. In this window, we can define the optical properties present in the model. Aluminum is used since the CubeSat is made of this material. Once the property has a name, the solar absorptivity and infrared emissivity can be defined. In addition, wavelength dependencies can be added, however, in this thesis radiation is assumed to be gray, diffuse.

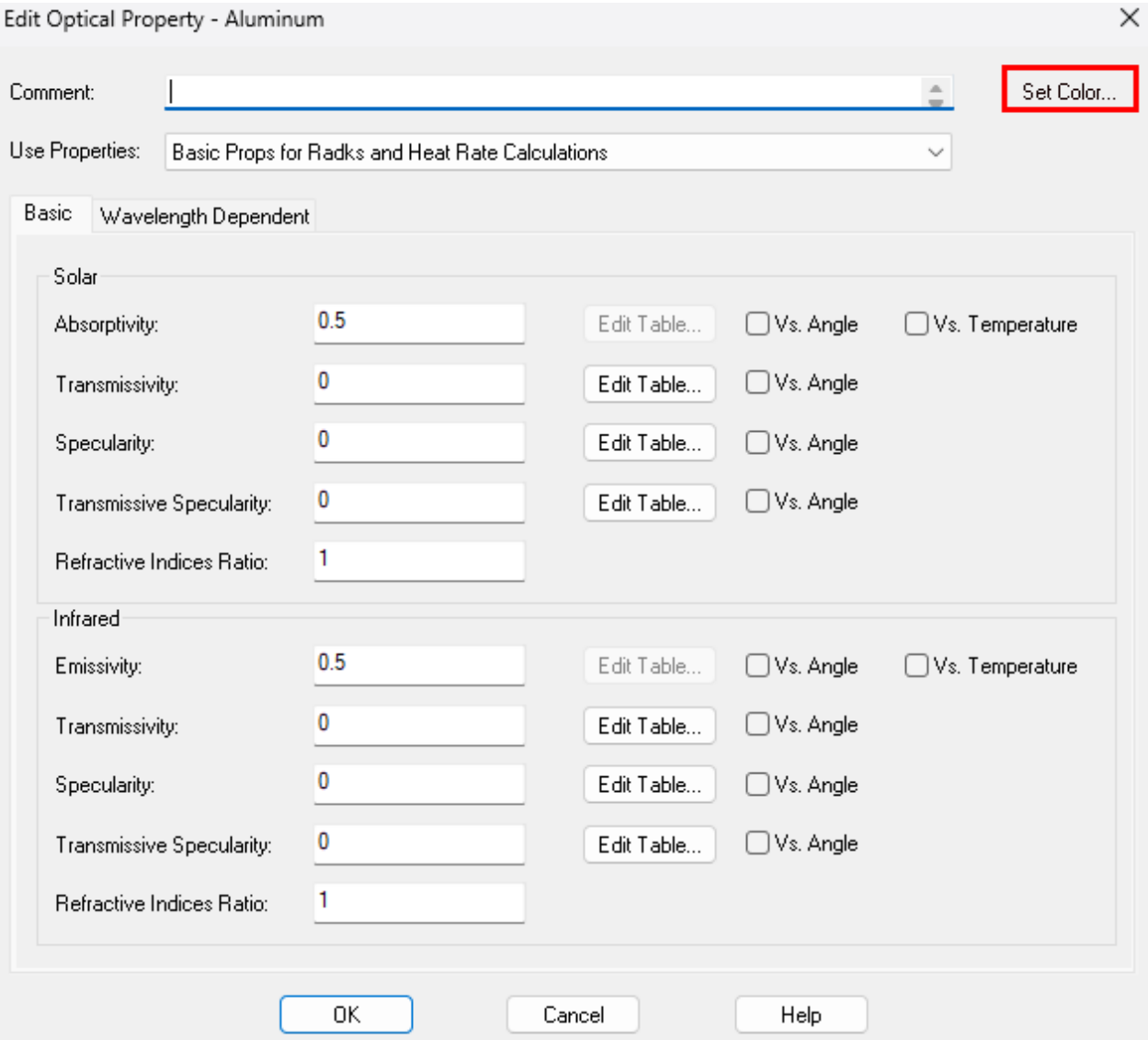

Figure 5.1: Thermal Desktop Optical Properties tab where the surface properties ϵ and γ can be defined.

[42]

Next, the thermal properties can be created by going to 'Thermal > Thermal Properties > Edit Property Data'. Again, we can create the name of the thermal property and then define the values for conduction, density, and specified heat.

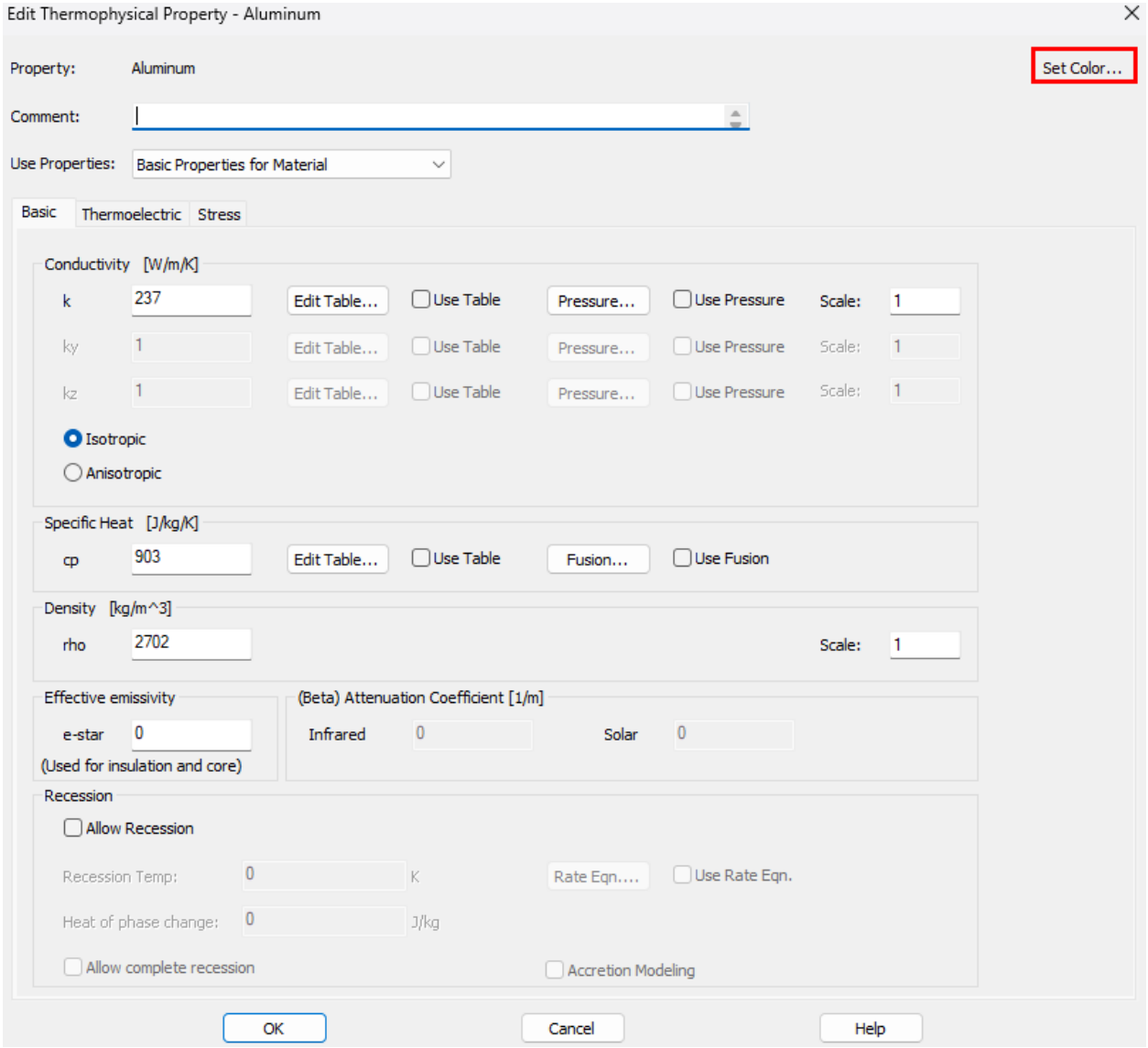

Figure 5.2: Thermal Desktop Thermal Properties tab where material properties like conductivity, specific heat, and density can be defined.
[42]

5.1.2 Creating Geometry

For this thesis, a fairly simple geometry is used. Since the enclosure has no energy storage, it will be modeled as a thin shell in Thermal Desktop. To do this select 'Thermal > Surfaces/Solids > Box' from the ribbon bar.

With the thin shell box created, the next task is to assign the thermal and optical properties to the box. This is done by double-left clicking on the box. A new window will appear, allowing us to edit the radiation and conduction properties of the box.

In the 'Radiation' tab, the 'Analysis Group Name' should be 'BASE', and the 'Active Side' should be 'both'. This ensures that the inside and outside of the box participate in radiation. Next, the optical properties are assigned to the box through a dropdown menu. To apply the properties change 'Default' to 'Aluminum'.

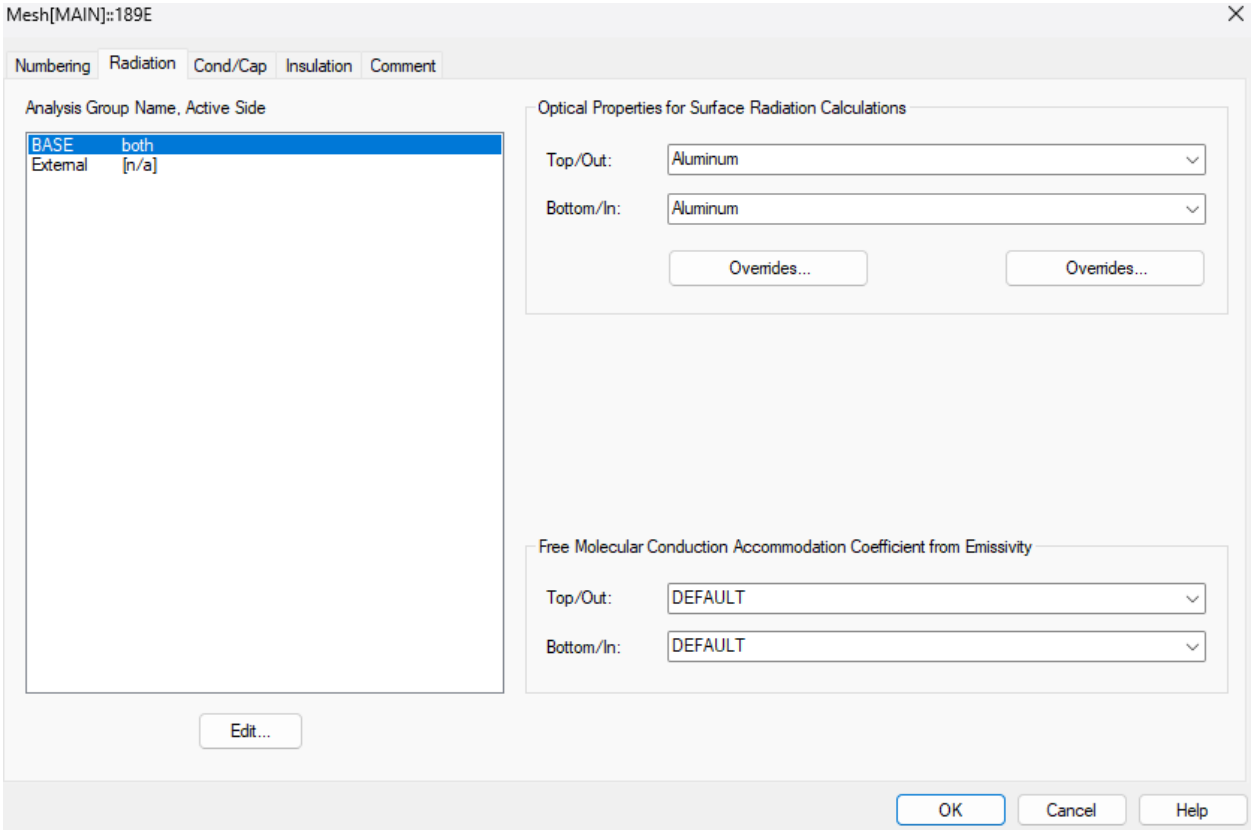

Figure 5.3: Radiation Tab used to assign the defined surface properties to the thin shell structure and define what surfaces can radiatively interact with each other.
[42]

Next, in the 'Cond/Cap' tab, select the material to be 'Aluminum'. Under the 'Gen Nodes' field, select 'As Arithmetic (Zero Capacitance)'. This ensures that the thin shell cannot store energy.

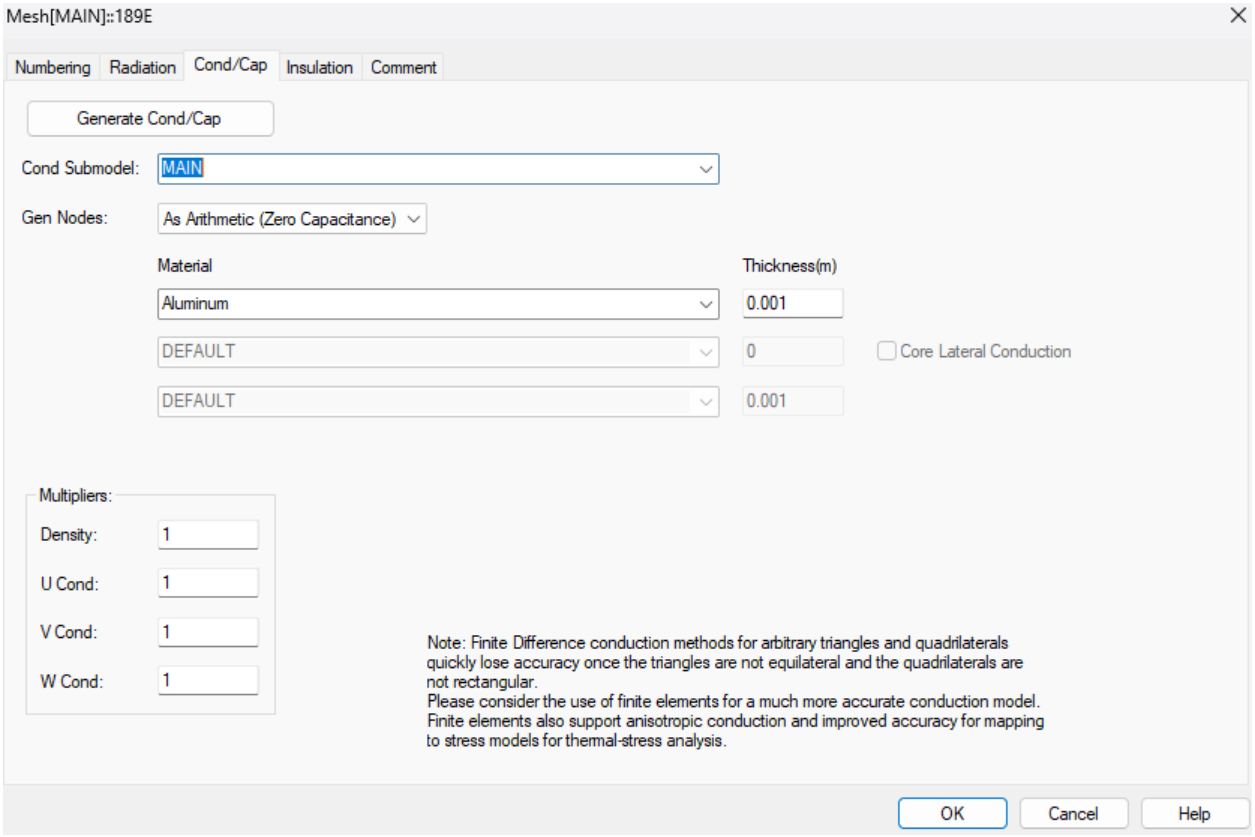

Figure 5.4: Conduction Tab used to assign material properties to the structure. [42]

Now it is time to define the optical payload. Since the payload can store energy it will be modeled as a finite difference solid. Select 'Thermal > Surfaces/Solids > Solid Brick'. Once this is done, a window will appear. In the 'Subdivision tab', ensure that center nodes are used and that the x,y, and z directions are set to 1. This will assign one node to the entire box. More nodes can be added by changing the values for each direction.

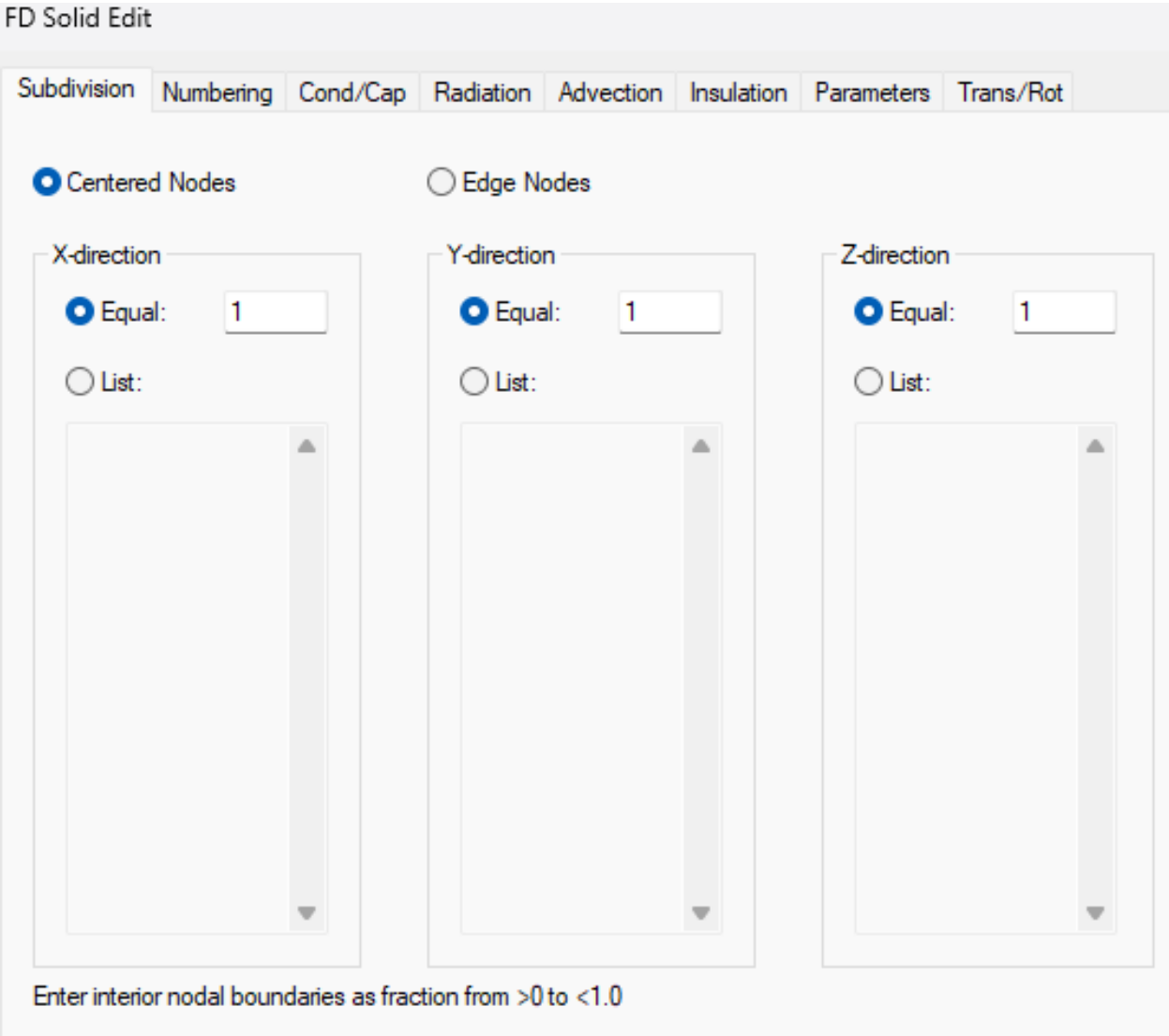

Figure 5.5: Subdivision Tab used to define how many nodes should represent the Optical structure. [42]

Next, under the 'Radiation' tab, the 'Analysis Groups' should be 'Outside' and applied to all faces. This makes sure that only the outside of the brick participates in the radiation exchange. Apply the defined optical properties to each surface in the same way as done for the box.

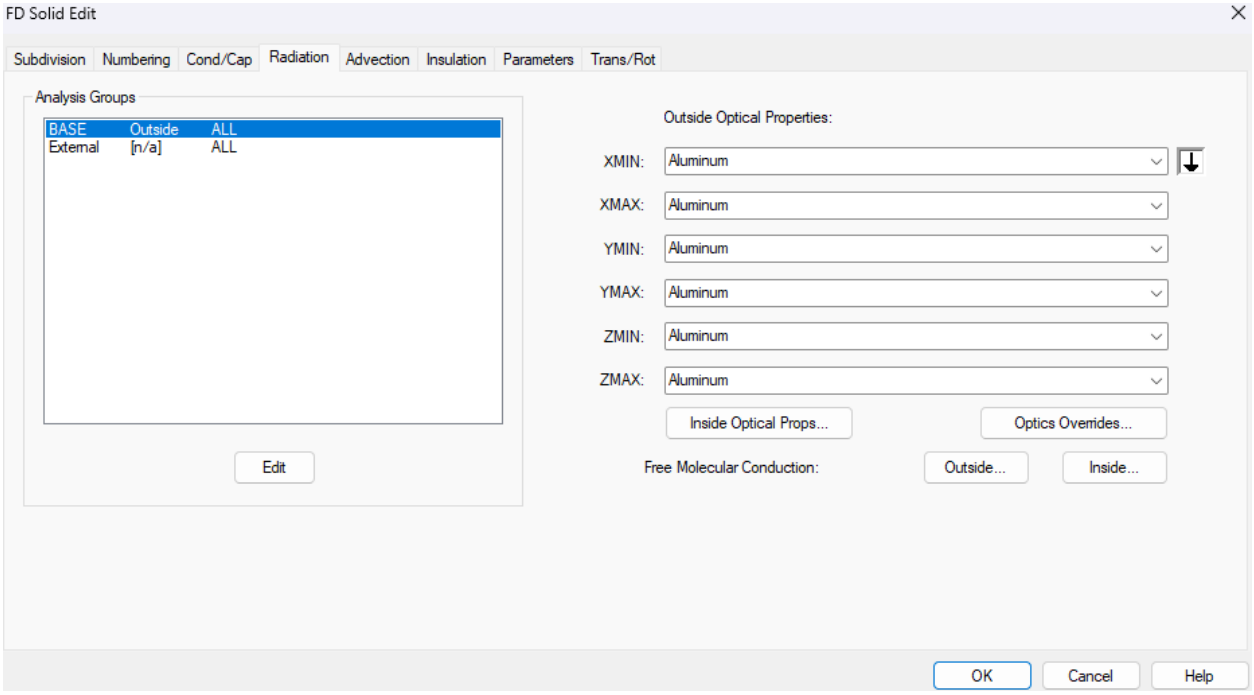

Figure 5.6: Radiation Tab used to assign the defined surface properties to the Optical system. [42]

Finally, under the 'Cond/Cap' tab, select the materiel to be the one defined previously and make sure that the 'Generate Nodes' field is set to 'Based on material property'. This will ensure that the optical system can store energy based on its material properties. Select OK and close the window.

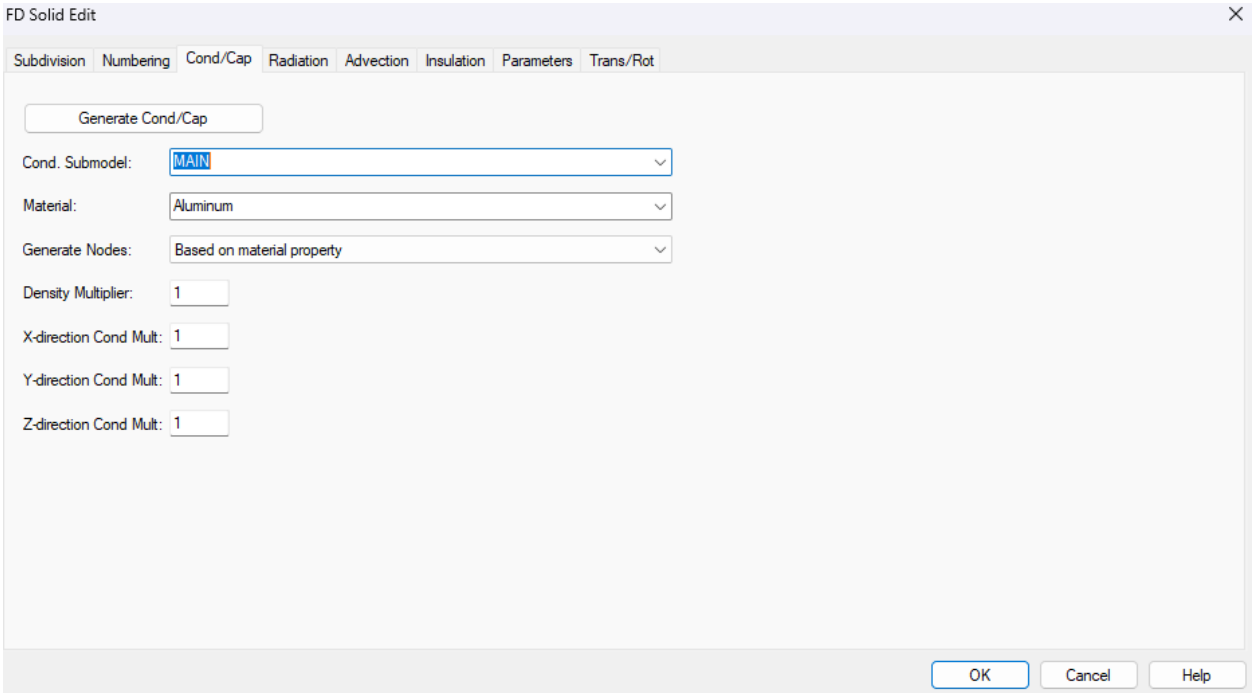

Figure 5.7: Conduction Tab used to set the material properties of the Optical system. [42]

Now that the geometry has been defined and properties assigned, it is time to run a simulation.

5.2 Running Simulations

To run a simulation, go to 'Thermal > Case Set Manager'. Double click 'Case Set 0' to modify the type of solution. In the 'Processes' field, check every box to make sure conduction and radiation are calculated and saved to your current working directory. In the 'Solution Type' field, deselect Steady State and check Transient.

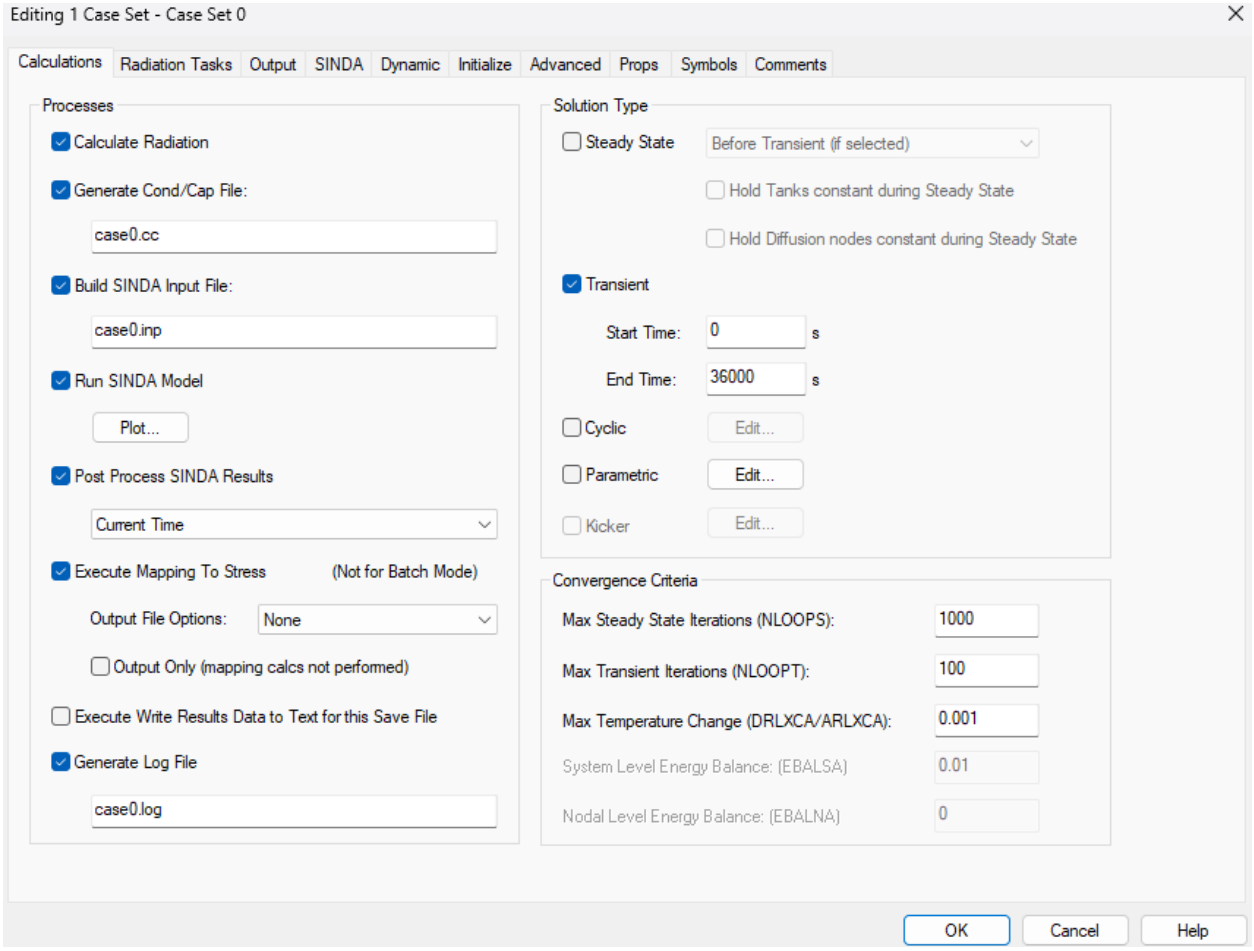

Figure 5.8: Case Set Calculation tab used to select the type of solution that should be performed. [42]

Next, select the 'Radiation Task' tab at the top. This tab is used to define how the radiation is calculated between each surface. Click 'Add' in the bottom left to define the radiation calculation type. Select 'View Factors' and then select the 'Advanced Control' tab. In this tab change the 'Nodalization Schemes' to 'Top/Bottom Specific'. Click OK to close this window. While still in the 'Radiation Task' tab, click 'Add' again. Set the 'Calculation Type' to 'Radks' and the 'Calculation Method' to 'Progressive Radiation'. Click OK to close the current window, and then click OK again to finish editing Case Set 0.

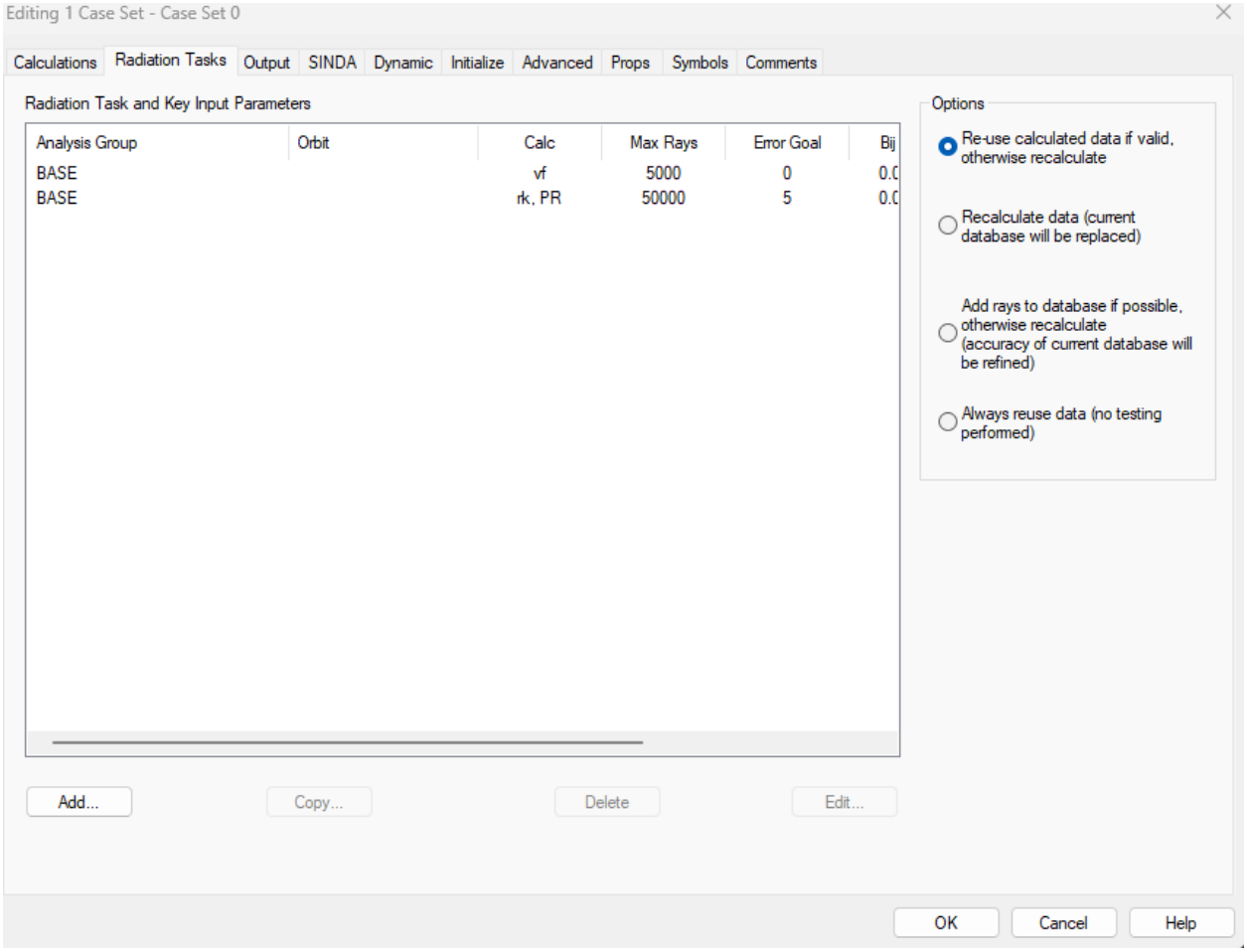

Figure 5.9: Case Set Radiation tab used to ensure a radiation analysis is conducted and that view factors are used instead of a ray trace.
[42]

5.3 Thermal Desktop Output

Select 'Case Set 0' in the 'Case Set Manager' then, in the bottom left, select 'Run 1 Selected Case'. The software will now run the solver and display the post-processing screen. Once complete, select the optical system by left clicking the brick once, then go to 'Thermal > Post Processing > X-Y Plot Data vs. Time'. This action will display the EZXY plotter, which will plot the temperature versus time for the optical system. This data can be exported to an excel file and imported to MATLAB by right-clicking the figure and selecting 'Export

Tabular Data'. This data can then be directly compared to the finite difference calculations done in MATLAB.

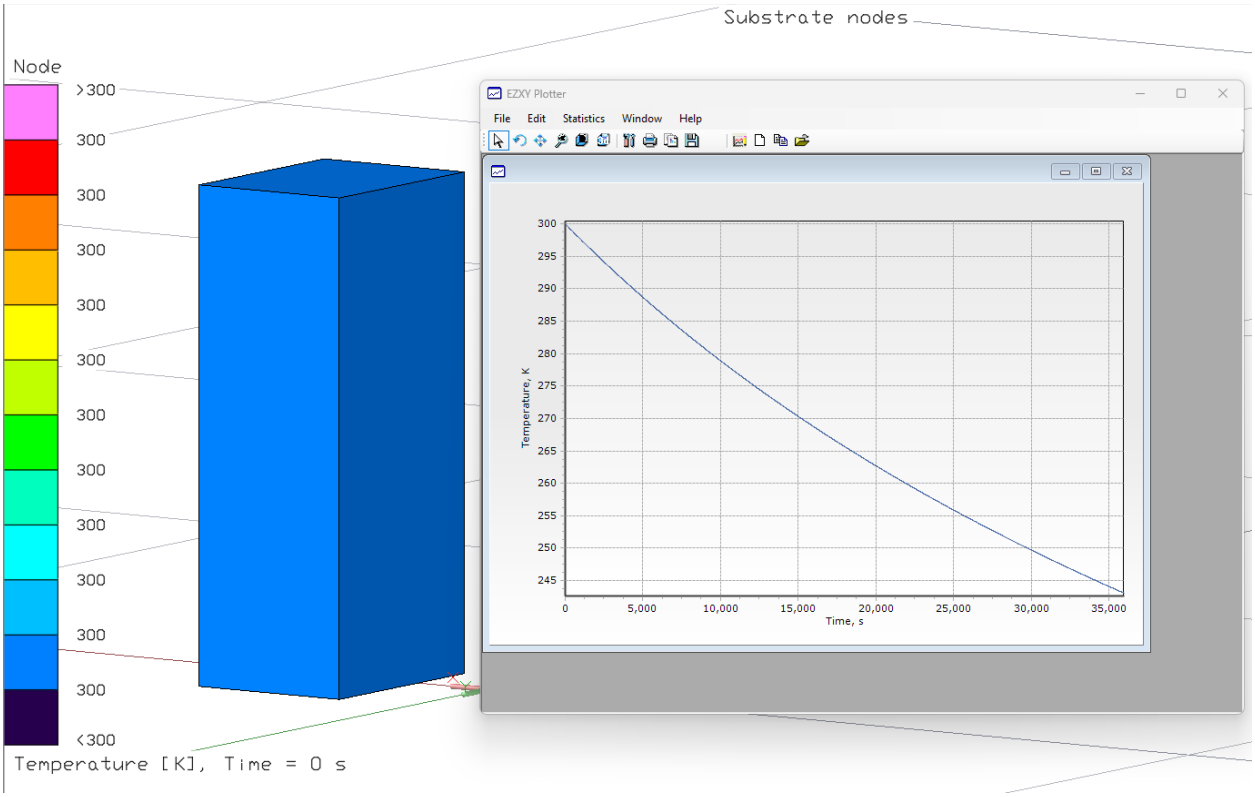

Figure 5.10: Optical payload temperature response with a initial condition of 300K. [42]

Chapter 6

Single Node Example

6.1 Single Node Thermal Modeling

The purpose of creating a single node model is to examine the feasibility of the methods derived in the previous chapters. Mainly, how to arrive at the A,B, and C matrices and then use a control law to control the temperature of the optical payload. To do this, we derive the nonlinear governing equation for the optical payload, convert the equation to its linear state-space form, and apply control laws that are designed to give a desired response.

Problem Statement: The optical payload (Orange) in Figure 6.1 is inside the CubeSat Structure (Cyan) and is initially at 300 Kelvin. The CubeSat radiatively loses heat to the surroundings which are at 2.73 Kelvin. Develop a controller to drive the CubeSats optics payload to 305K with no steady-state error. Assume all structures are made of aluminum with absorptivity and emissivity equal to 0.5. The CubeSat is discretized into two nodes each with their own control volumes. Node one is used to represent the entire optical payload, while node two is used to represent the CubeSats structure.

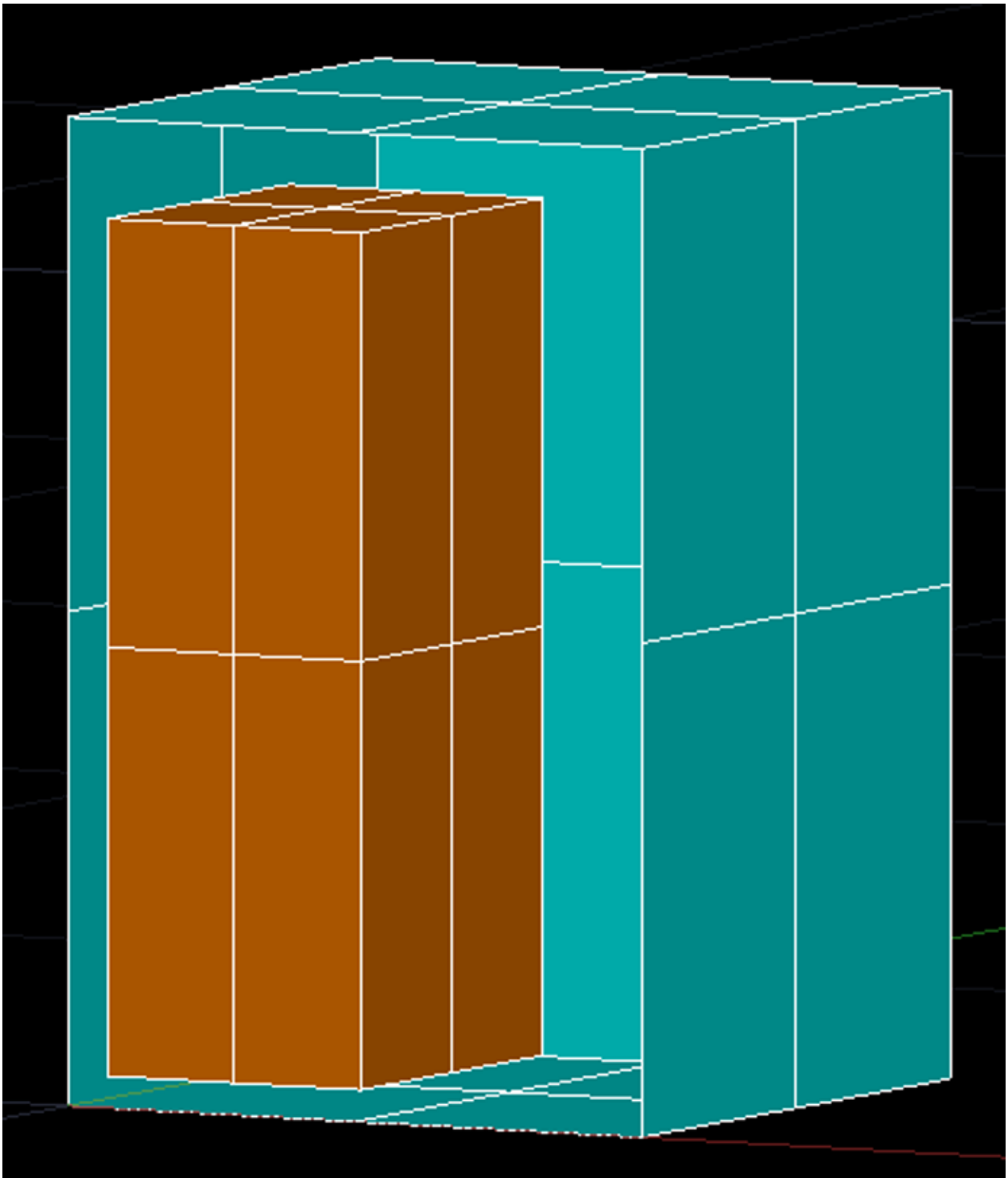

Figure 6.1: Thermal Desktop CubeSat depiction - Thin shell structure (cyan) completely encloses the optical payload (orange). A face from the structure is made invisible so that we can see the internal contents of the CubeSat.

The first task is to derive the nonlinear governing equation for the optical payload. This is done with equation 4.3. Since the entire optical payload is represented by a single node, there is no conduction, as this requires the optical payload to be represented by at least two nodes. Because of the absence of conduction, the only mode of heat transfer that the optical payload experiences is radiation. Therefore, the energy balance for the interior of the CubeSat is described by equation 4.1,

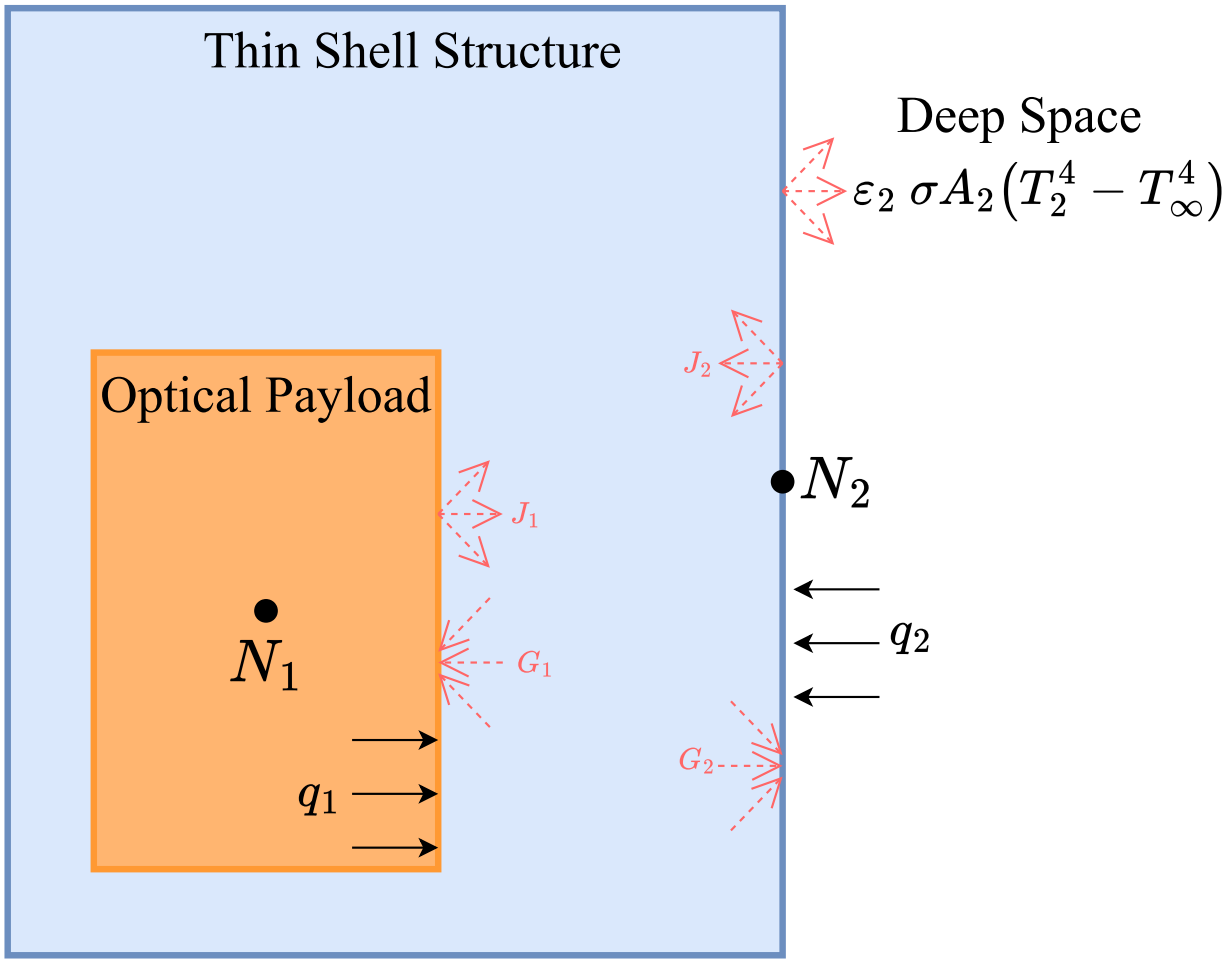

Figure 6.2: CubeSat energy balance for Figure 6.1 depicting the radiative interaction between the two isothermal surfaces. It is important to note the q_2 is actually defined to be inside the the structures surface as depicted in Figure 6.3.

Since there are only two surfaces representing the enclosure, the optical payload and the thin shell structure, equation 4.1 becomes the following system of equations,

$$\left(\frac{\delta_{1,1}}{\epsilon_1} - F_{1,1}\frac{1-\epsilon_1}{\epsilon_1}\right)\frac{q_1}{A_1} + \left(\frac{\delta_{1,2}}{\epsilon_2} - F_{1,2}\frac{1-\epsilon_2}{\epsilon_2}\right)\frac{q_2}{A_2} = (\delta_{1,1} - F_{1,1})\sigma T_1^4 + (\delta_{1,2} - F_{1,2})\sigma T_2^4, \tag{6.1}$$

$$\left(\frac{\delta_{2,1}}{\epsilon_1} - F_{2,1}\frac{1-\epsilon_1}{\epsilon_1}\right)\frac{q_1}{A_1} + \left(\frac{\delta_{2,2}}{\epsilon_2} - F_{2,2}\frac{1-\epsilon_2}{\epsilon_2}\right)\frac{q_2}{A_2} = (\delta_{2,1} - F_{2,1})\sigma T_1^4 + (\delta_{2,2} - F_{2,2})\sigma T_2^4. \tag{6.2}$$

The view factors can be found through view factor algebra which takes advantage of the fact that the view factor from any given surface to all other surfaces summed together in a enclosure always equals one [41, Ch. 4]. Since the optical payload (surface 1) is inside the structure (surface 2), $F_{1,1} + F_{1,2} = 1$. Because surface 1 cannot see itself, $F_{1,1} = 0$ and therefore $F_{1,2} = 1$. $F_{2,1}$ can be found from the reciprocity relationship $A_1F_{1,2} = A_2F_{2,1}$, which results in $F_{2,1} = \frac{A_1}{A_2}$. Since it is assumed that only the temperature of the optical payload is known, the above system of equations has two equations with three unknowns. To remove a unknown and solve the system of equations, we will need an extra equation. The third equation can found by observing the radiation outside the CubeSats structure. By assuming the enclosure is a thin shell and taking special care to define properties consistent with the internal control volume one gets,

$$\dot{E}_{in} = 0,$$

$$-q_2 - \epsilon_2\sigma A_2(T_2^4 - T_\infty^4) = 0.$$

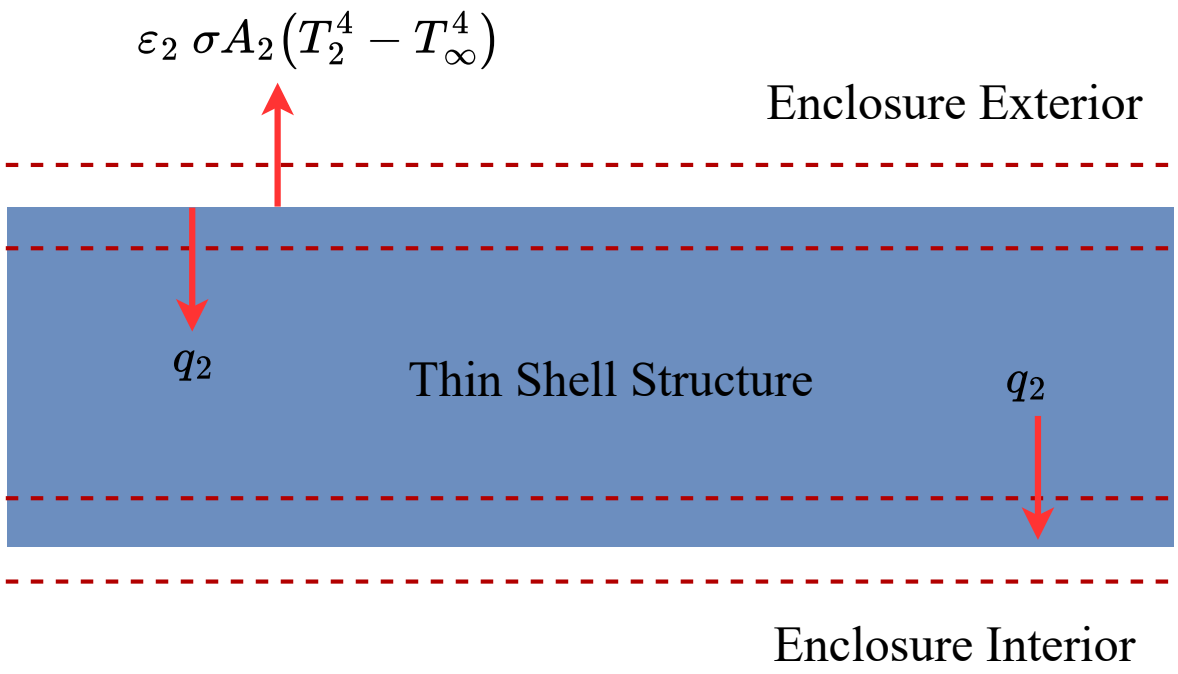

Figure 6.3: Thin shell structure control volume used to link the CubeSats interior radiation to the CubeSats exterior ration. q_2 is defined in a manor to stay consistent with the enclosure theory assumptions.

With these three equations, the net radiation loss of the optical payload can be found in terms of its temperature,

$$q_1^p = \frac{(T_1^p)^4 - 55.55}{3.39 \times 10^8} \text{ (Watts)}. \tag{6.3}$$

Since $q_1(T)$ is defined as the instantaneous net radiation loss of the optical payload, the finite difference equation becomes,

$$\begin{aligned} \dot{E}_{st} &= \dot{E}_{in} + \dot{E}_{gen} \\ \rho c_p V \frac{T_1^{p+1} - T_1^p}{\Delta t} &= -q_1^p + \dot{E}_{gen}^p \\ T_1^{p+1} &= -\Delta t \gamma q_1^p + \Delta t \gamma \dot{E}_{gen}^p + T_1^p. \end{aligned} \tag{6.4}$$

Using,

$$\rho = 2702 \text{ (kg/m}^2\text{)}, \quad c_p = 903 \text{ (J/kg/K)}, \tag{6.5}$$

$$\epsilon = 0.5, \quad A_1 = 0.01664 \text{ (m}^2\text{)}, \tag{6.6}$$

$$A_2 = 0.4156 \text{ (m}^2\text{)}, \quad T_1 = 300 \text{ (K)}, \tag{6.7}$$

$$T_\infty = 2.74 \text{ (K)}, \quad \dot{E}^p_{gen} = 0 \text{ (W)}, \tag{6.8}$$

and solving for q_1^p and then solving for T_1^{p+1} at each time step gives

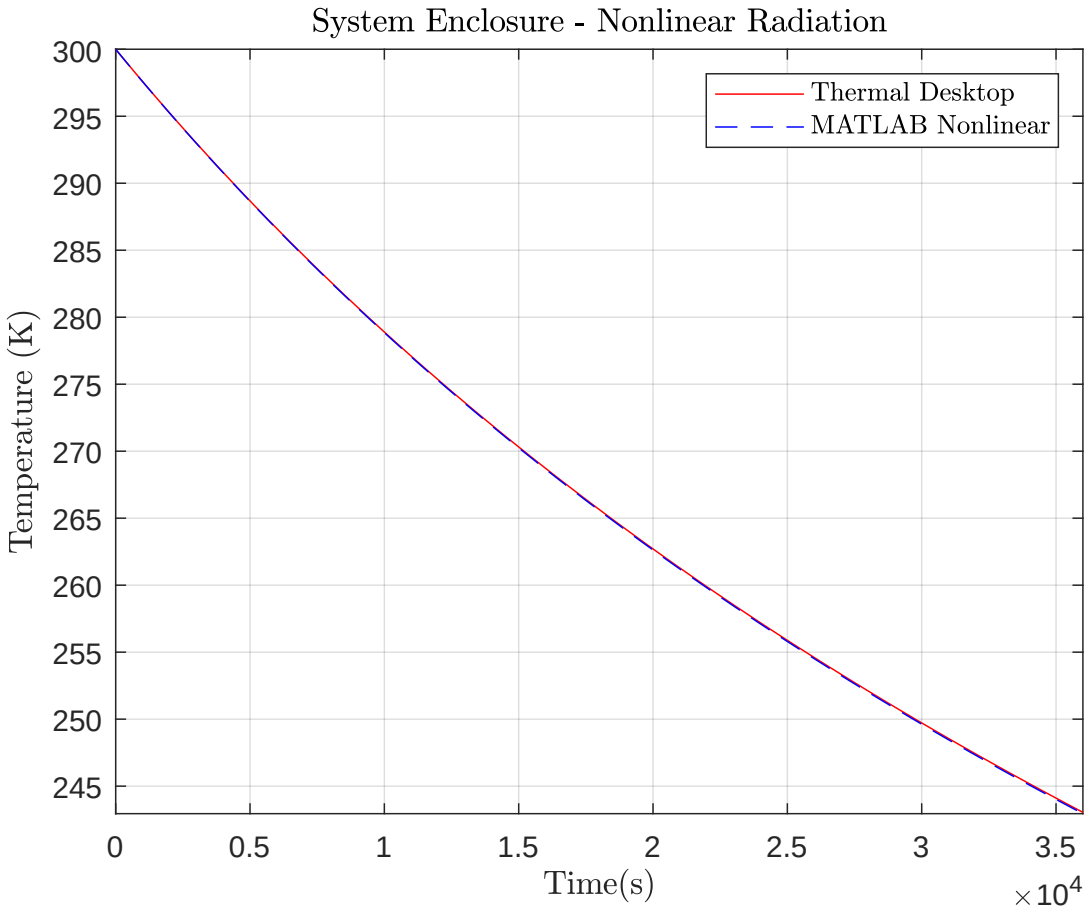

Figure 6.4: Derived nonlinear single node temperature response compared to Thermal Desktop results. The agreement between MATLAB and Thermal Desktop results validates the methodology used.

From the above figure, it is clear that the nonlinear finite difference equation matches the Thermal Desktop results which validates the thermal modeling method taken. By viewing the residuals between MATLAB and Thermal Desktop by taking the absolute value of their difference, we can quantify this agreement.

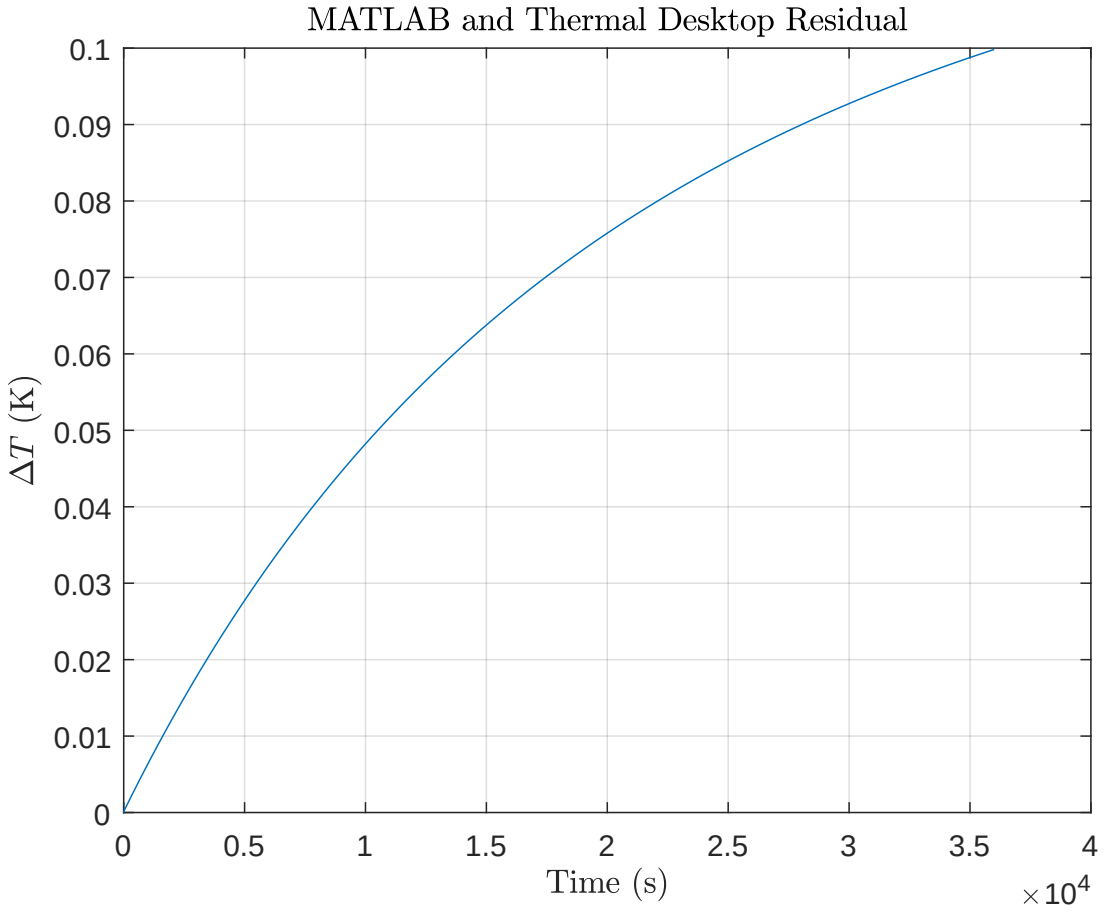

Figure 6.5: Single node residual analysis between MATLAB and Thermal desktop. These residuals validate the approach taken as after 10 hours the difference between results is only 0.1 Kelvin

To reach a linear state-space model for the optical payload, the nonlinear radiation term needs to be linearized with respect to temperature about the equilibrium point ($T_{eq} = 300K$,

$u_{eq} = \dot{E}_{eq,gen} = 23.87W$) as done in Section 2.6. Doing so gives,

$$\Delta\mathbf{T}_{k+1} = [0.999]\Delta\mathbf{T}_k + [0.0001]\Delta\mathbf{u}_k, \tag{6.9}$$

which results in Figure 6.6 after setting $\Delta\dot{E}_{gen} = \Delta\mathbf{u}_k = -23.87W$ and by adding T_{eq} to ΔT to get the unperturbed temperature free response. Note the change of notation from T^p to T_k, where the superscript p is changed to the subscript k. So for the future if k is used as a subscript it represents the current time step. If k is seen alone then is represents the thermal conductivity.

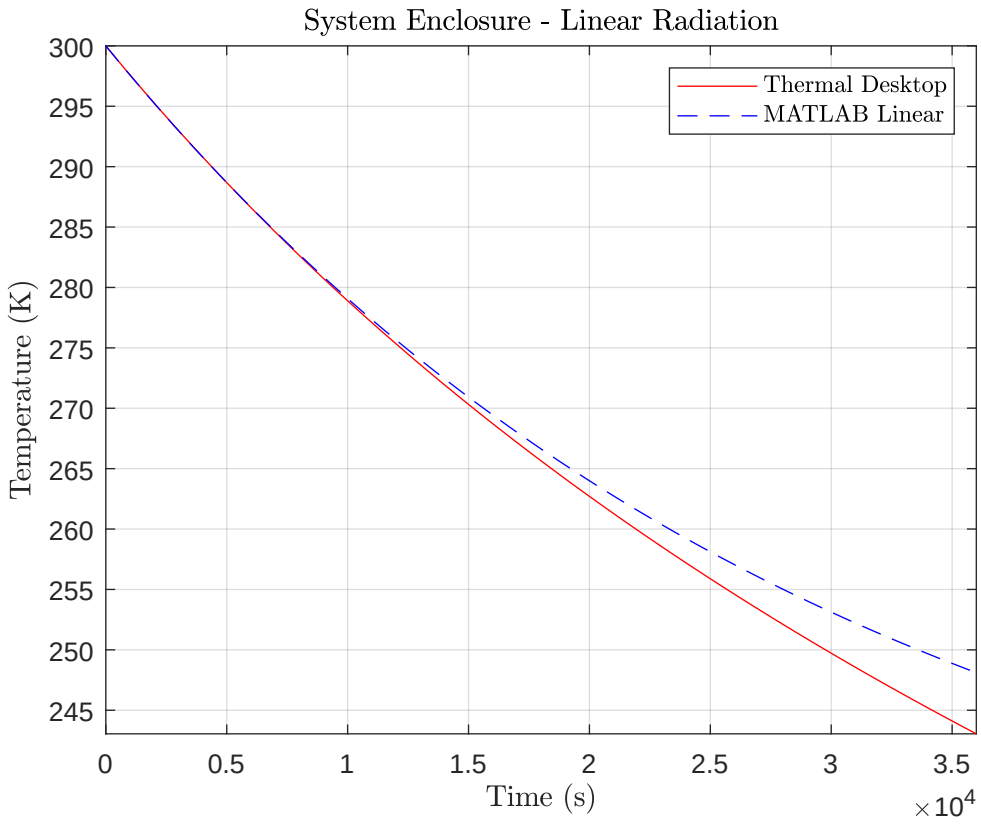

Figure 6.6: Single node linear temperature response compared to the nonlinear Thermal Desktop results. Note that the linear response is linear in temperature, which is why the linear response does not show a linear relationship when plotted against time.

As expected, the linear response matches well with the nonlinear response around the operating point, meaning that there exists a region where the nonlinear and linear governing equations are identical to each other. This is a key result as even though we plan to control the linear dynamics, we are actually controlling the true nonlinear dynamics if we stay near the operating point. Another notable result is that the linearized radiation equation is linear with respect to temperature and not time. Because of this, the blue dashed line in Figure 6.6 is nonlinear with respect to time.

6.2 Single Node Thermal Control

From equation 6.9, it is seen that the open-loop system has stable dynamics since the eigenvalues of A are inside the unit circle. This is not surprising as heat transfer is known to have a stable response towards its equilibrium point.

To control this system and close the loop, the following control law is used,

$$\mathbf{\Delta u}_k = G\mathbf{\Delta r}_k - K\mathbf{\Delta T}_k, \tag{6.10}$$

where $\Delta\mathbf{r}_k$ is the change in the reference signal. The closed-loop system then becomes,

$$\mathbf{\Delta T}_{k+1} = (A - BK)\mathbf{\Delta T}_k + BG\mathbf{\Delta r}_k, \tag{6.11}$$

where K is found by placing the poles of A at $\lambda = 0.999$. Finally, G is found through the equation 2.31.

The closed-loop dynamics are now given by,

$$\mathbf{\Delta T}_{k+1} = 0.999\mathbf{\Delta T}_k + 0.001\Delta \mathbf{r}_k, \tag{6.12}$$

$$\Delta \mathbf{y}_k = \Delta \mathbf{T}_k, \tag{6.13}$$

assuming that the state (ΔT_k) can be observed with a sensor. If the change in the reference signal is a step input that steps from 300K to 305K at 0 seconds, then the optical payloads temperature will track the step signal with zero steady-state error, as in Figure 6.7.

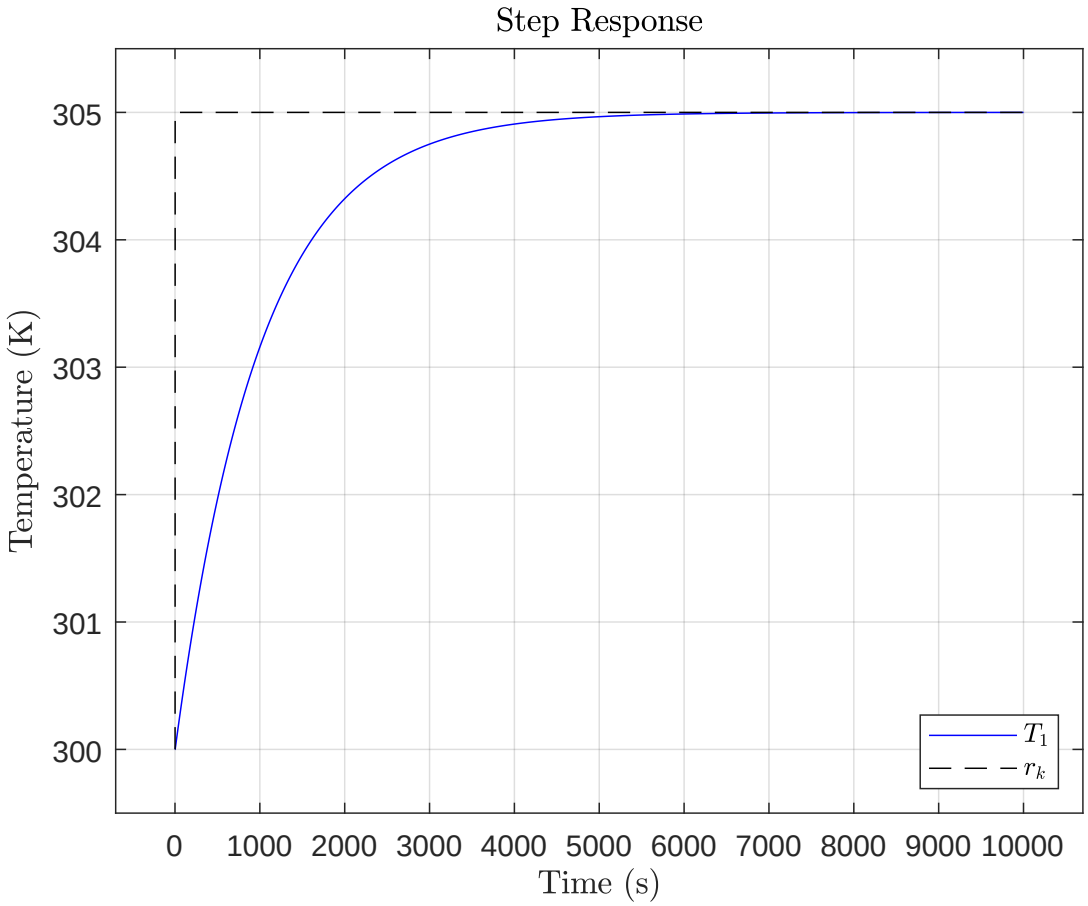

Figure 6.7: Single node temperature response after closing the loop and implementing state feedback with steady-state tracking and shifting the temperature by T_{eq}. Note that the transient response is dictated completely by the designers pole placement choice.

The required input to achieve this is found by back substituting $G, K, \mathbf{\Delta r}_k$ and $\mathbf{\Delta T}_k$ into the control law. The required input can then be plotted against time as done in Figure 6.8.

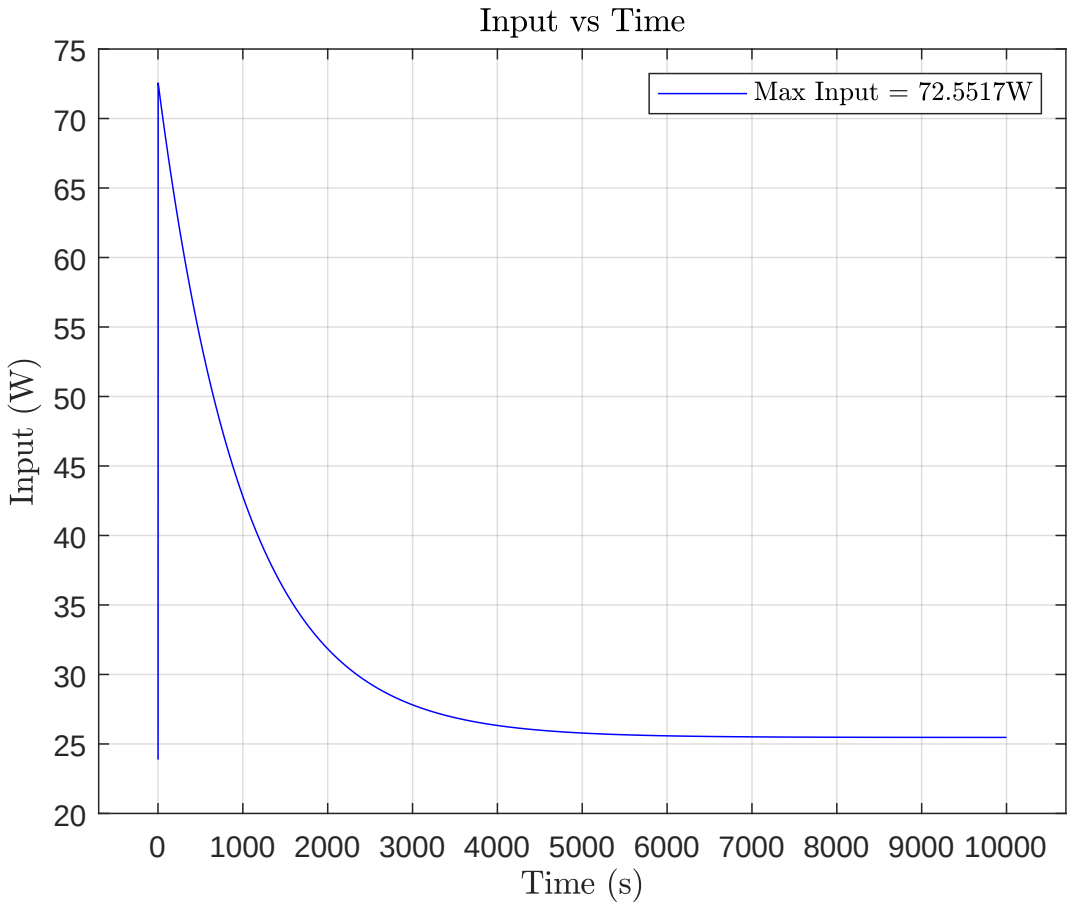

Figure 6.8: Input required to produce the temperature response in Figure 6.7 after shifting by u_{eq}. A large input peak occurs as the control law, $\mathbf{\Delta u}_k = G\mathbf{\Delta r}_k - K\mathbf{\Delta T}_k$, initially produces larges values since there is a significant difference between $\mathbf{r}_k$ and $\mathbf{x}_k$. The Input then settles to its steady-state value as $\mathbf{r}_k$ and $\mathbf{x}_k$ become closer.

From the above figure, it is observed that a max input of 72.6 Watts and then a steady-state input of 25.5 Watts is required to regulate the optical payload from 300K to 305K as done in Figure 6.7. For the current CubeSat thermal configuration, this max input exceeds what the CubeSat can generate from solar-panels alone. Since the solar radiative flux around Earth is on average 1350 W/m^2, and the cross sectional area for where a solar panel could be

attached to the CubeSats structure is 0.0512 m^2, the total incident energy on the solar panel is 69.12 Watts. Next, taking into account that the efficiency of a solar panel is about 30%, the total energy collected by the panel is 20.7 Watts. This means that the CubeSat would be in a energy deficit its entire life even if in constant sun light. A feasible next step would be to change the thermal and radiative properties of the CubeSat so that the steady-state heating value of 20.7 Watts can be reached. The max input is not worrisome, as this value can be reduced by changing the close loop eigenvalues.

The importance of eigenvalue placement can be seen in Figures 6.7 and 6.8 as this is what dictates the response in each figure. If λ were placed closer to 1, the system would respond slower and the max input would be smaller. The opposite is true if λ were placed closer to 0. Additionally, for systems with more than one input, if λ were chosen to be complex conjugates of each other, the system would be second order, as some overshoot would exist in the response. However, since precise temperature regulation is desired, complex conjugate poles should be avoided.

A minor inconvenience of the approach taken is that there are no constraints on the input. For example, if the initial state was 300K but the desired final state is 3,000K, there would exist an input that would drive the system to that final temperature. To avoid these unfeasible inputs, input constraints would need to be included. However, pole placement is not as useful in this context as the poles cannot be arbitrarily placed when constraints exist. In addition, pole placement focuses on tailoring the system response, not a constrained search. These reasons violate the fundamental concept of pole-placement; therefore, it is important for the designer to understand the limits of their system.

Since a linear model is used to control the nonlinear dynamics, the derived control law can only be used in a region where the linear and nonlinear models agree well with each other. This region was identified as 280K to 320K from Figure 6.6, within which the difference between the linear and nonlinear models remains below 0.25K. If the optical payload exceeds this temperature region, then a new controller will need to be made for the new initial conditions and operating point. The same derived thermal model can be reused with updated values to determine a new control law. The CubeSat could then switch between control laws depending on its current temperature.

Chapter 7

Eight Node Model

7.1 Eight Inputs Eight Outputs

A eight node model is a step up in complexity from the one node model as conduction is now considered. The finite difference equation is of the exact same form as equation 4.3. This leads to very similar results to the one node case. The benefit of the 8 node case is that now there are up to 8 inputs that can control the temperature of the optics box.

Problem Statement: The optics payload (Orange) in Figure 6.1 is inside the CubeSat Structure (Cyan) and is initially at 300 Kelvin. The CubeSat radiatively loses heat to the surroundings at 2.73 Kelvin. Develop a controller to drive the optics box to 305 Kelvin with no steady-state error. Assume all structures are made of aluminum with absorptivity and emissivity equal to 0.5. The CubeSat is discretized into nine nodes with their own control volumes. The optical payload is represented by nodes one through eight, and the CubeSat structure is represented by node nine.

As in the single node case, we want to derive the finite difference equations for the optics box. To model the radiation inside the CubeSat, equation 4.1 should be used, and it is reiterated for convenience below.

$$\sum_{j=1}^{N}\left(\frac{\delta_{kj}}{\epsilon_j} - F_{kj}\frac{1-\epsilon_j}{\epsilon_j}\right)\frac{q_j}{A_j} = \sum_{j=1}^{N}(\delta_{kj} - F_{kj})\sigma T_j^4 \tag{7.1}$$

The unknown view factors can be solved using view factor algebra and reciprocity relationships [41, Ch. 4]. This task is made easier by remembering that each surface on the optics box cannot see itself (surfaces 1 - 8). This leads to a view factor matrix,

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.5998 \end{bmatrix}. \tag{7.2}$$

Again, this system of equations has 9 equations with 10 unknowns since only the temperature of nodes one through eight are known. The extra equation needed to solve this system is found by observing the radiation outside the CubeSats thin shell structure. The extra equation needed is found to be the same as the single node case,

$$-q_2 - \epsilon_9\sigma A_9(T_9^4 - T_\infty^4) = 0 \tag{7.3}$$

With these 10 equations, q^p_{1-8} can be found in terms of its current temperature T^p_{1-8}.

These derived equations can be combined with equation 4.3 to arrive at a complete thermal representation of the optical payload.

$$\begin{aligned} T_i^{(p+1)} = {} & \gamma k A_x \Delta t \left[\frac{T^p_{-x} + T^p_{+x} - 2T^p_i}{\Delta x} \right] \\ & + \gamma k A_y \Delta t \left[\frac{T^p_{-y} + T^p_{+y} - 2T^p_i}{\Delta y} \right] \\ & + \gamma k A_z \Delta t \left[\frac{T^p_{-z} + T^p_{+z} - 2T^p_i}{\Delta z} \right] \\ & + \gamma \Delta t \dot{E}^p_{gen} - \gamma \Delta t q^p_{i,rad} + T^p_i \end{aligned} \tag{7.4}$$

By identifying the nodes surrounding node N_i and inserting the known values for $q^p_{i,rad}$, a nonlinear equation for each node in the optical system can be found. The temperature profile for each node in this equation can be determined by iteratively stepping through time as seen in Figure 7.1.

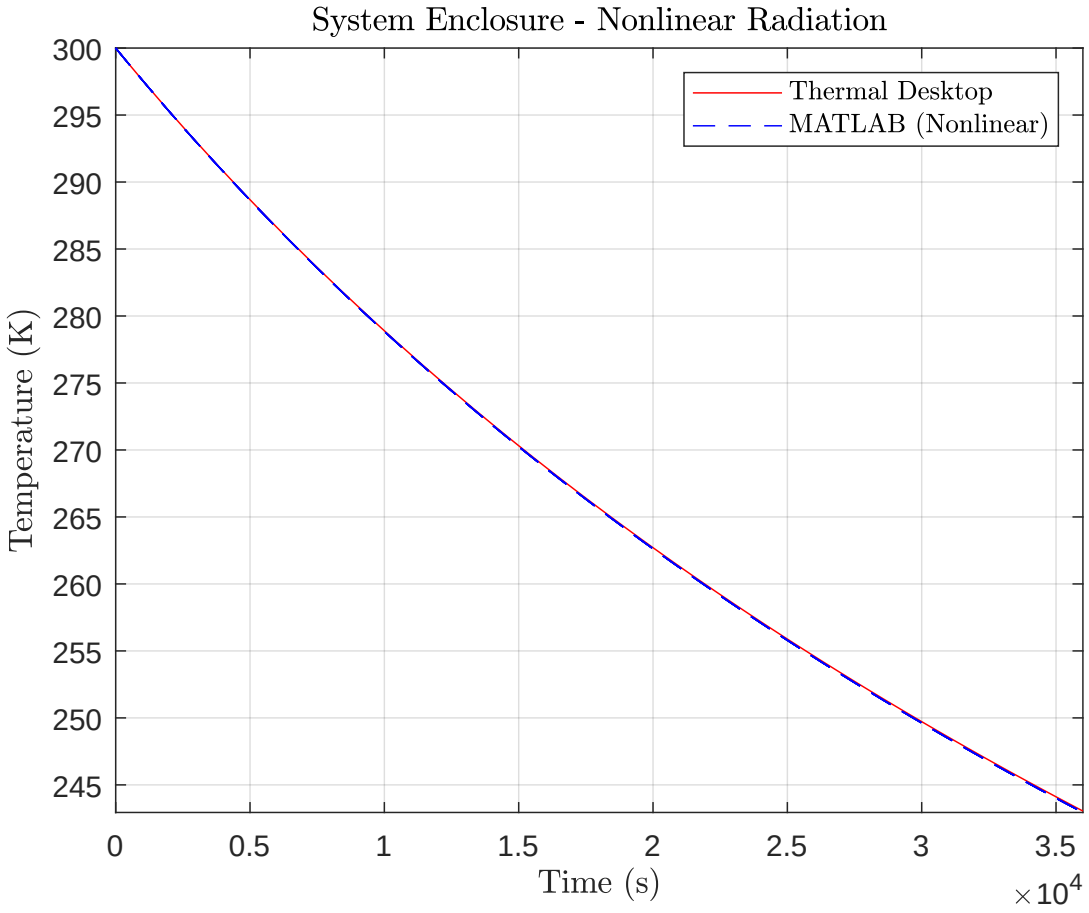

Figure 7.1: Derived nonlinear eight node temperature response compared to Thermal Desktop results. The agreement between MATLAB and Thermal Desktop results validates the methodology used.

Since each node in the thermal model starts at a initial equilibrium temperature of 300K and experiences the same radiation from the CubeSats structure, the nonlinear temperature profile of each of the eight nodes is similar to that of the one node case. The initial conditions can be changed to show that conduction has been successfully applied. Figure 7.2 illustrates that the primary mode of heat transfer for the CubeSat is through radiation, as conduction is highly damped. For this reason, it is not surprising that the single node and eight node temperature profiles match when starting at the same initial conditions.

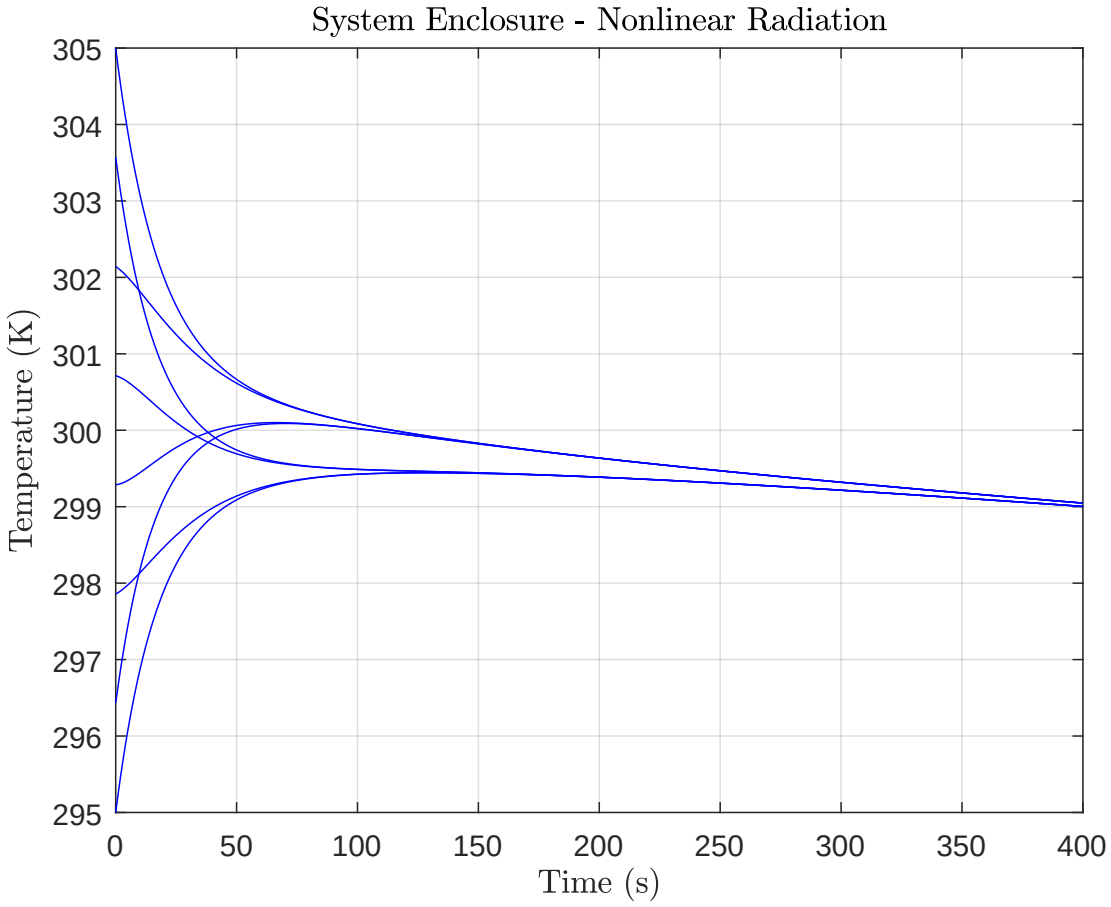

Figure 7.2: Derived nonlinear eight node temperature response with different initial conditions. Used to illustrate that conduction is highly damped.

Again, the agreement between results in Figure 7.1 validates the approach taken. By taking the difference between the Thermal Desktop and MATLAB results, we can quantify this agreement in Figure 7.3.

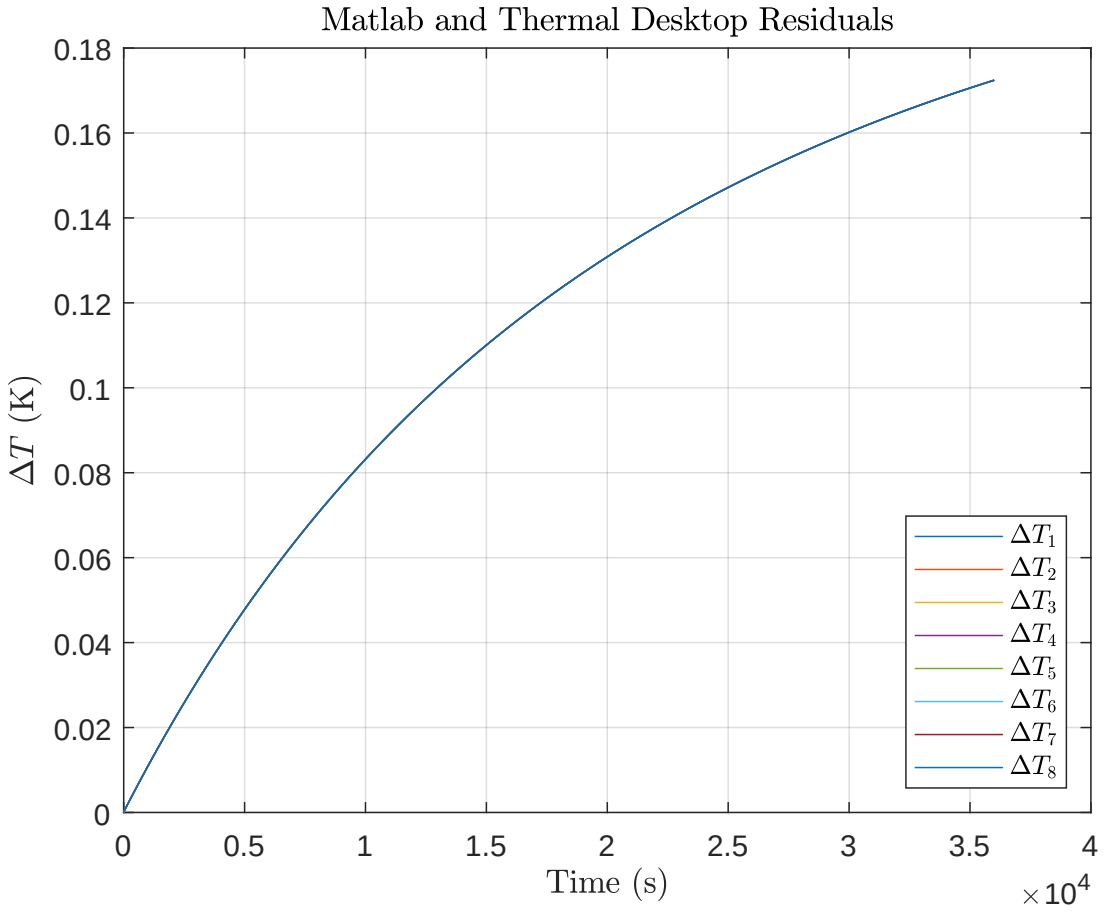

Figure 7.3: Residual analysis between Thermal Desktop and MATLAB. The small residuals indicate that there is practically no difference between results on this time scale. At the end of a 10 hour simulation, the results differ by less than 0.18 Kelvin.

To emphasize that Figures 7.1 and 7.3 actually have eight lines representing the temperature profiles of each of the eight nodes of the optical system, we can zoom in on the residuals as illustrated by Figure 7.4.

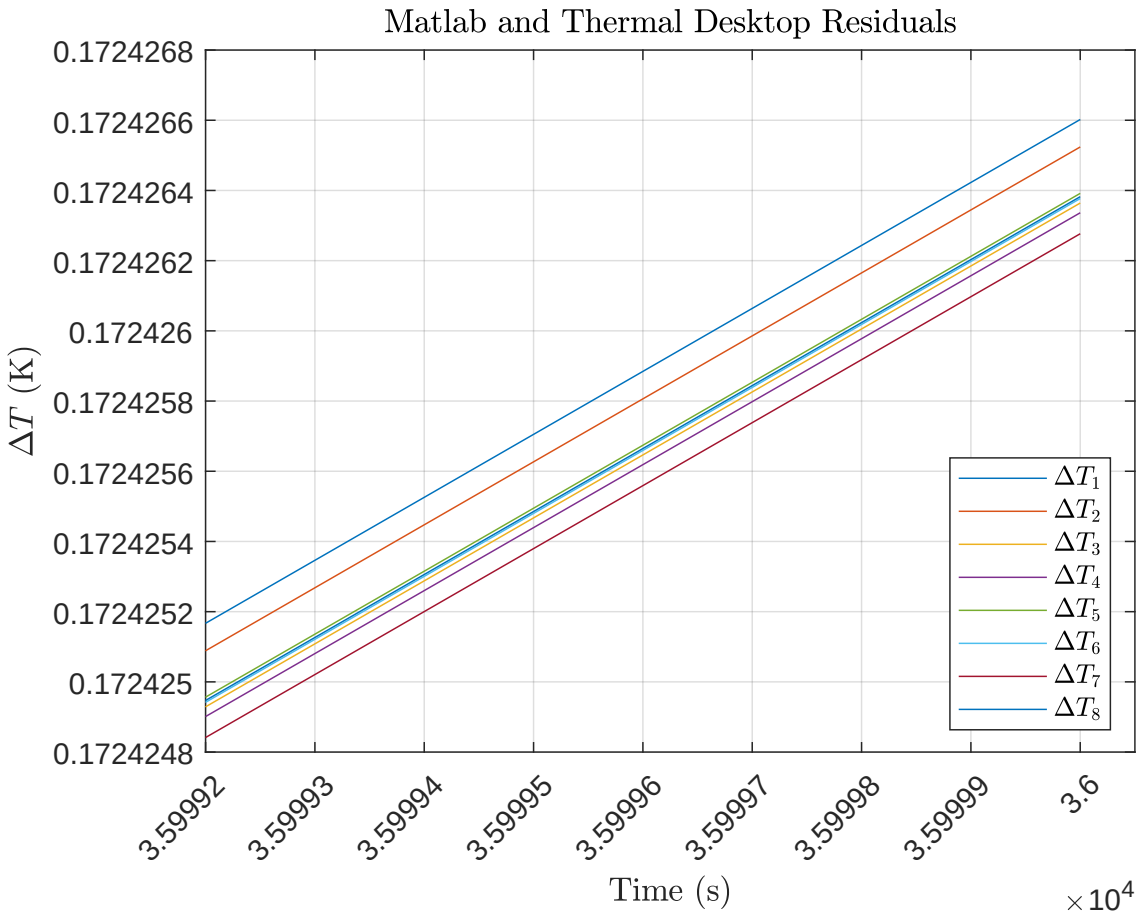

Figure 7.4: Magnified version of Figure 7.3 and used to illustrate that there are in fact eight temperature profiles being plotted in Figures 7.1 and 7.3.

The results from Figure 7.1 are nonlinear and in order to find a linear system the radiation terms need to be linearized. This can be done with a Taylor series as done in Section 2.6,

$$\Delta\mathbf{T}_{k+1} = A\Delta\mathbf{T}_k + B\Delta\mathbf{u}_k, \tag{7.5}$$

where $A \in \mathbb{R}^{8\times8}$, $B \in \mathbb{R}^{8\times8}$, and $\Delta\mathbf{T}_{k+1}, \Delta\mathbf{T}_k, \Delta\mathbf{u}_k \in \mathbb{R}^{8\times1}$; The expansion point of the Taylor series is $T_{eq} \in \mathbb{R}^{8\times1} = 300K$ and $u_{eq} \in \mathbb{R}^{8\times1} = 2.98W$ for each of the eight nodes.

Again, solving for the linear temperature free response in the same way as the single node case leads to Figure 7.5.

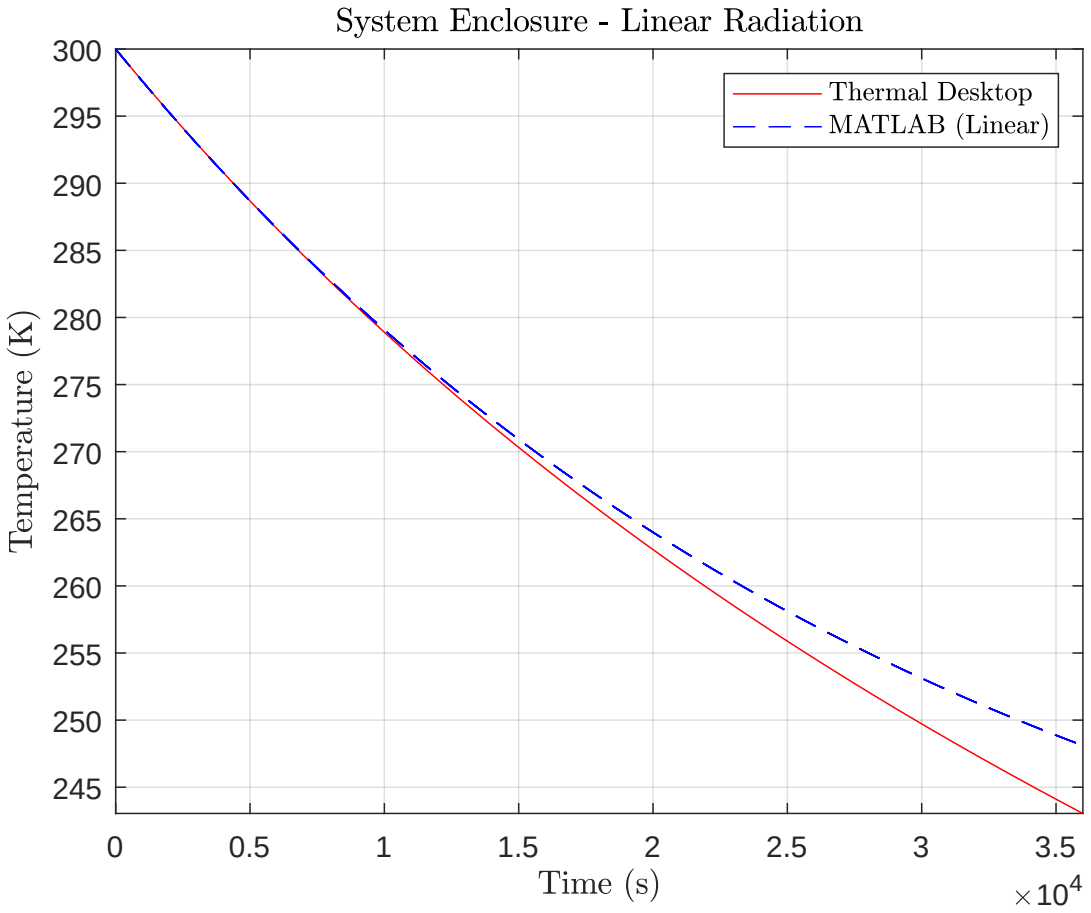

Figure 7.5: Linear eight node temperature response compared to the nonlinear Thermal Desktop results.

Next, the same control law from the one node case can be used to perform state feedback with steady-state tracking.

$$\boldsymbol{\Delta}\mathbf{u}_k = G\boldsymbol{\Delta}\mathbf{r}_k - K\boldsymbol{\Delta}\mathbf{x}_k, \tag{7.6}$$

with K and $G \in \mathbb{R}^{8\times 8}$.

K is found by placing the 8 poles of the system. The closed-loop poles are arbitrarily chosen as $\lambda = [0.7, 0.73, 0.76, 0.8, 0.83, 0.86, 0.88, 0.89]$. G is found by substituting in known values into equation 2.31.

The closed-loop system can now be formed in the same fashion as the single node case. Doing so leads to Figure 7.6.

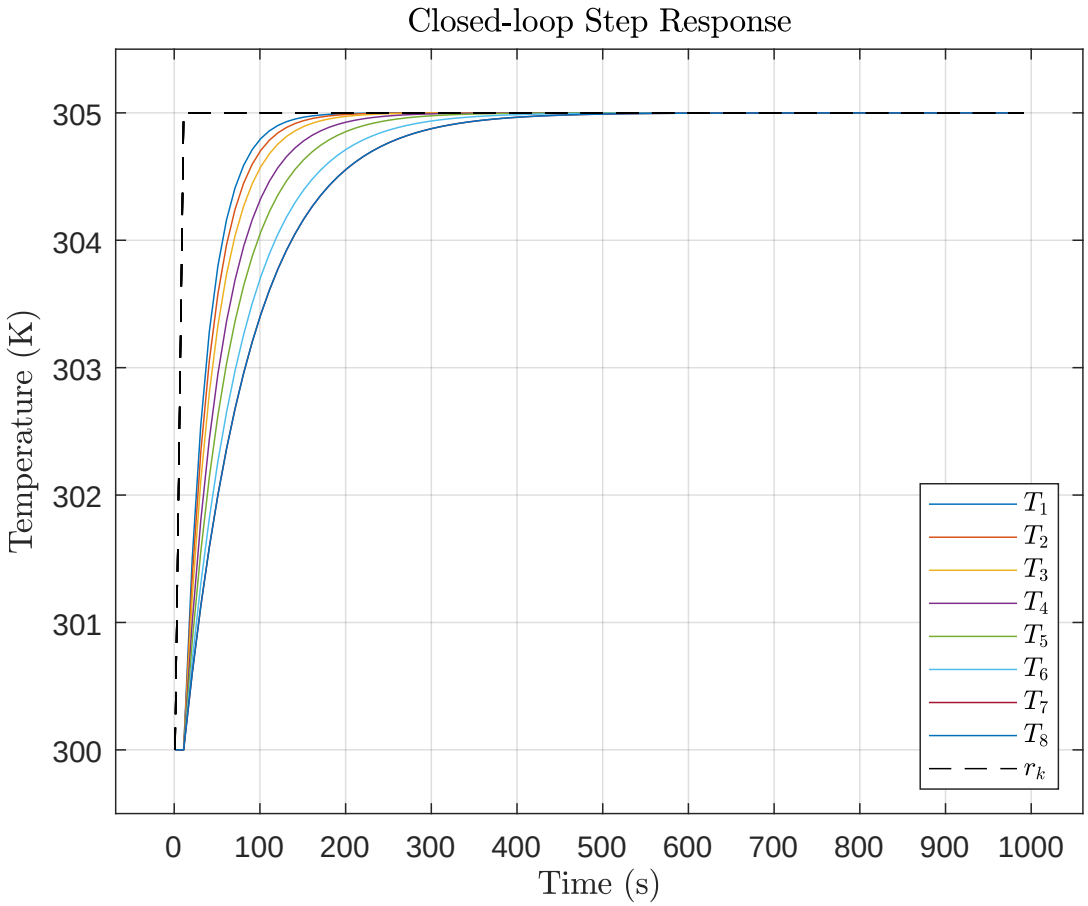

Figure 7.6: Eight node closed loop step response illustrating the effects of pole placement. Poles with magnitude close to zero have a quick response and poles with magnitudes close to one responded slower.

As can be seen, the eight node system tracks the reference signal with no steady-state error as designed. Since each node has a different eigenvalue, each node has its own response. The

nodes with eigenvalues closer to one respond slower, and the nodes with eigenvalues closer to zero respond faster. The input response to produce this output is given in Figure 7.7.

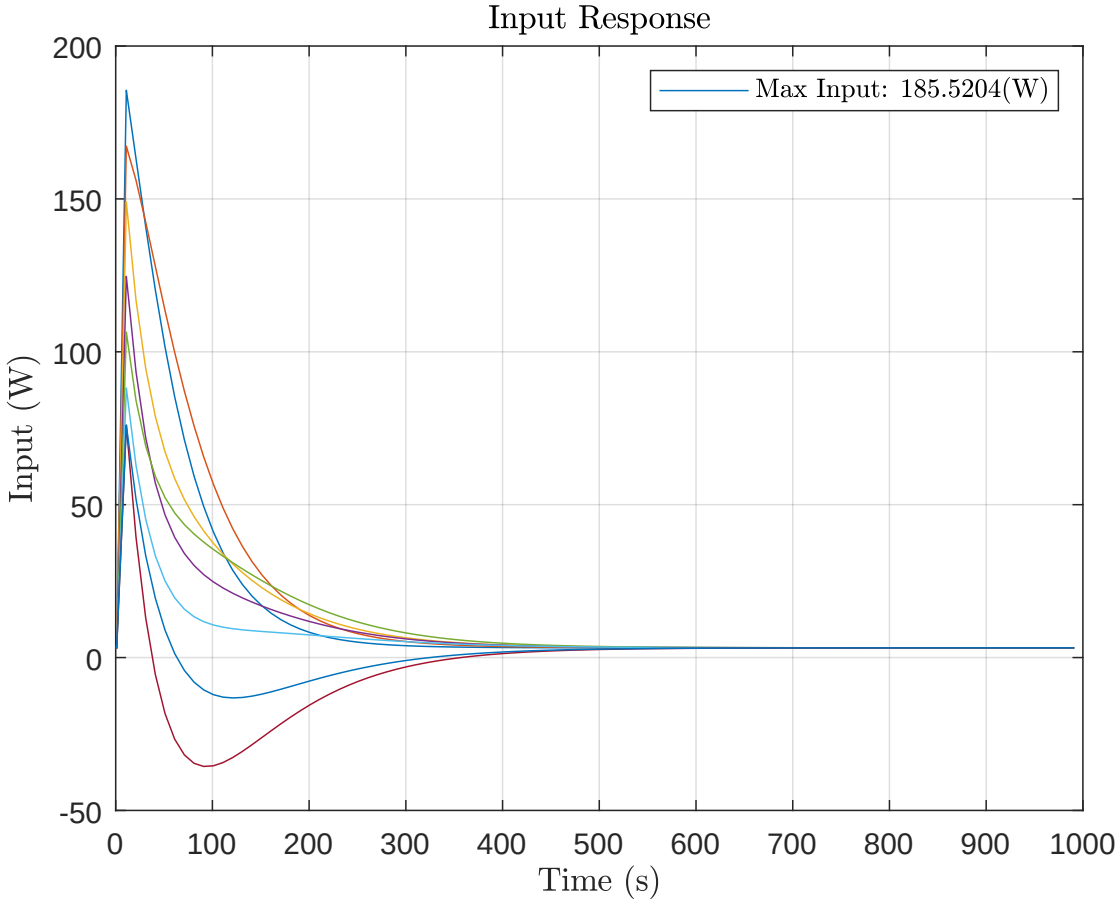

Figure 7.7: Input required to move the optical system as done in Figure 7.6. A large input peak occurs as we are quickly changing the temperature of the optical system by 5 Kelvin (over 1000 seconds).

Although an eight-input eight-output thermal model and control law for the optical payload is convenient to show that the linear control method taken works, it is not a system likely to be encountered in practice. There will exist nodes to which it is not possible to physically attach heaters. Therefore, in the next section we provide an analysis for a model that only includes seven-inputs and eight-outputs.

7.2 Seven Inputs Eight Outputs

As the number of nodes increases, there will be nodes where it is physically impossible to attach heaters. This lack of inputs will cause the B matrix to no longer be square and, in turn, potentially prevent perfect steady-state tracking. To see this effect, we take the original eight node system, and remove a column from its B matrix. This modifies the system into one where seven of the eight nodes have an input and all the nodes have a sensor that outputs their temperatures.

To control this system we use a similar approach to what we have done previously. In the previous approaches, we linearized the dynamics to form a perturbed state-space equation, like equation 6.9. However, now that B is not square, we need to take special care when finding the steady-state matrix G.

To start our analysis we start with the perturbed dynamics,

$$\Delta \mathbf{T}_{k+1} = A\Delta \mathbf{T}_k + B\Delta \mathbf{u}_k. \tag{7.7}$$

The output can also be written is the same form,

$$\Delta \mathbf{y}_k = C\Delta \mathbf{T}_k, \tag{7.8}$$

where $A \in \mathbb{R}^{8\times 8}$, $B \in \mathbb{R}^{8\times 7}$ and $C \in \mathbb{R}^{8\times 8}$. Our equilibrium vectors are chosen to be,

$$\mathbf{T}_{eq} = \begin{bmatrix} 300.0975 \\ 300.0328 \\ 300.0970 \\ 300.0268 \\ 300.0953 \\ 300.0147 \\ 300.0940 \\ 299.9864 \end{bmatrix} (\mathrm{K}), \mathbf{u}_{eq} = \begin{bmatrix} 3.415 \\ 3.415 \\ 3.415 \\ 3.415 \\ 3.415 \\ 3.415 \\ 3.415 \end{bmatrix} (\mathrm{W}). \tag{7.9}$$

The equilibrium vectors (T_{eq}, u_{eq}) were determined from the derived nonlinear dynamics, and chosen to be close to $300K$. Because one of the nodes does not have a heating input, the equilibrium vectors are non-constant.

From here we can design and control the dynamics in the usual fashion with,

$$\Delta \mathbf{u}_k = G\Delta \mathbf{r}_k - K\Delta \mathbf{T}_k, \tag{7.10}$$

which results in,

$$\Delta \mathbf{T}_{k+1} = (A - BK)\Delta \mathbf{T}_k + (BG)\Delta \mathbf{r}_k \tag{7.11}$$

$$\Delta \mathbf{y}_k = C\Delta \mathbf{T}_k, \tag{7.12}$$

where K and $G \in \mathbb{R}^{7\times 8}$.

The gain K is designed by placing the eight poles at $\lambda = [0.8, 0.81, 0.82, 0.84, 0.85, 0.87, 0.88, 0.90]$. In order to design the gain G, equation 2.35 is used which will provide steady-state tracking if $\min_G ||I - VG||^2 = 0$ which is the case when V has a right inverse. If $\min_G ||I - VG||^2 \neq 0$, then we have approximate steady state tracking, which is the case for this example. Finally, the perturbed reference signal ($\Delta \mathbf{r}_k$) to be tracked is a step from 0 to 5 Kelvin for each node. Plugging the designed K and G into the closed-loop system, simulating the perturbed temperature and input responses, and finally shifting the perturbed temperature and input response by the equilibrium point ($\mathbf{T}_{eq}, \mathbf{u}_{eq}$) results in figures 7.8 and 7.9. As seen in Figure 7.8, our control design tracks the reference signal extremely well, however there are small steady-state errors that persist.

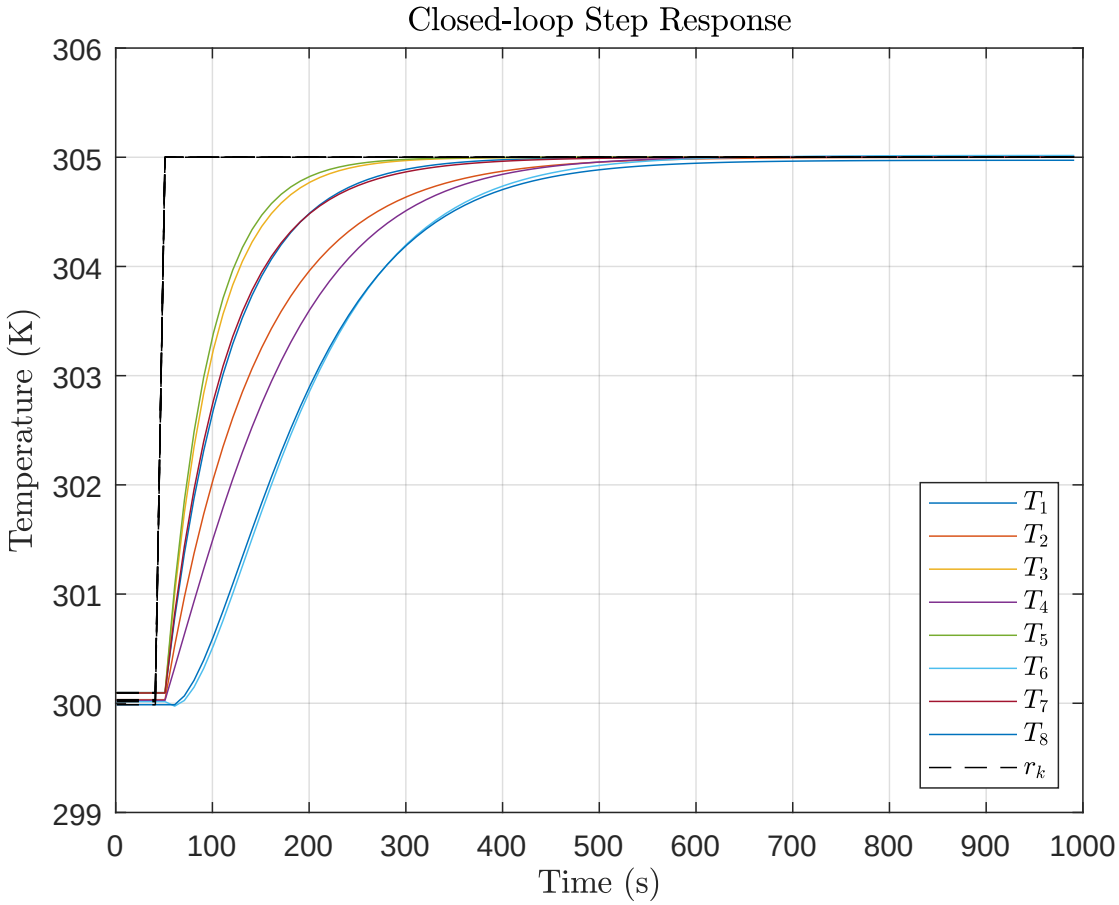

Figure 7.8: Closed loop step response illustrating the steady-state error that is caused by only using seven heaters.

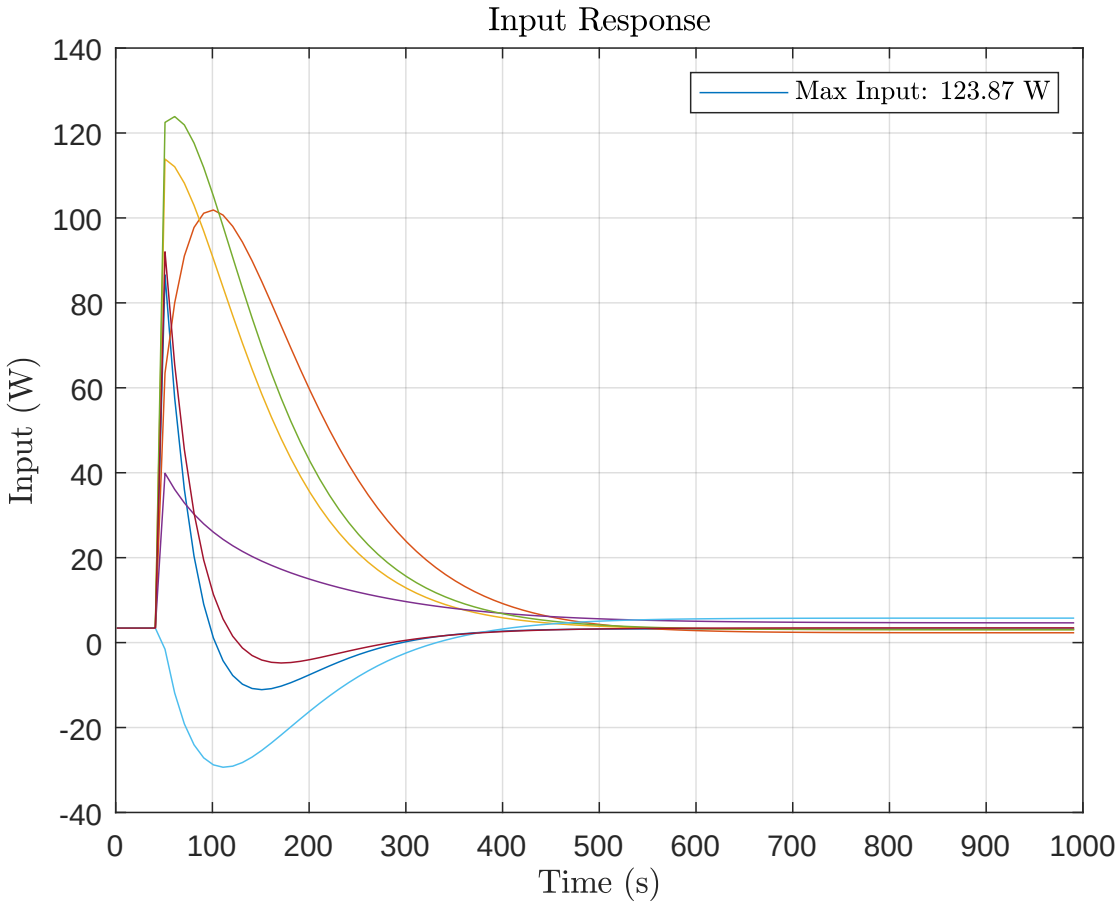

Figure 7.9: Input required to move the optical system as done in Figure 7.8.

Chapter 8

Summary & Future Work

The motivation behind this thesis is to maintain a CubeSats optical payload at a predetermined temperature to ensure that the geometric relationship between optical components remains as designed. The optical payload represents all optical components, including the primary and secondary mirrors of the telescope. If the mirrors are not maintained at their designed temperatures, thermal expansion will change the geometric relationship between the two mirrors, leading to blurry science images. Therefore, a stable thermal environment that does not change the geometrical properties of the optical payload is necessary. To achieve this goal, we first discretize the CubeSat into elements and create a nonlinear finite difference equation for the optical payload that accounts for; 1. Conduction between elements in the optical payload; 2. Radiation between the elements of the optical payload, the structure surrounding the optical payload, and deep space; 3. Heating source inside each optical payload element that is used to increase its temperature if needed. This nonlinear finite difference model is then linearized and written in a linear state-space form for which linear control methodologies can be applied to ensure the optical system remains at its designed temperature. The main methodology we take advantage of is the control law, $\mathbf{u}_k = G\mathbf{r}_k - K\mathbf{x}_k$, which provides precise tracking of the reference signal through the proper

design of the gain matrices G and K. By setting the tracking signal equal to the optical payload design temperature, we can ensure that the optical payloads geometrical relationships remain as designed.

The approach outlined was carried out for two different CubeSat discretizations; 1. One element representing the optical payload and one element representing the CubeSat structure surrounding the optical payload; 2. Eight elements representing the optical payload and one element representing the CubeSat structure surrounding the optical payload. In both element discretizations, we were able to derive a nonlinear finite difference equation that matched with a high-fidelity thermal simulator which verifies our thermal modeling approach. We next linearized the finite difference equations to arrive at a linear state-space equation for both CubeSat discretizations. By assuming that all the elements in the optical payload could have their temperatures controlled by their heating sources, and that each element has a sensor to output the elements temperature, we were able to design a controller for the optical payload. This controller maintains each element of the optical payload at its designed temperature to ensure the geometrical relationships are unaltered.

Another controller was designed for the eight optical payload element case, but this time with a different configuration. In this new configuration we assume that only seven of the eight elements in the optical payload have a heating source, and that all the elements have a sensor to output their temperature. By doing this, we explore the feasibility of maintaining the optical payload at its designed temperature despite a portion of the payload not having a heating source. It was found that a controller does exists that can control the system as desired, however it will not provide steady-state tracking. This controller will have some steady-state error due to necessary modifications to the G matrix in the control

law, $\mathbf{u}_k = G\mathbf{r}_k - K\mathbf{x}_k$. Even though this controller will have some issues with maintaining steady-state tracking, as long as the temperature of the optical payload is known and close to its designed temperature, optical engineers can make adjustments to produce a clear image.

In short, this thesis illustrates how to model and control the temperature of a CubeSat in space. However, the current controller design and results are based purely in theory, so a logical next step would be to test the controller design and validate the results with a physical system. Fortunately, this is possible in the University of Arizona Space Astrophysics Lab, as we have access to a thermal vacuum chamber that can simulate space environments. Another next step would be to create finer models with more nodes. With more discretizations comes more accuracy and the ability to simulate realistic heating source locations. Finally, another direction we could head in is using a different control method to regulate the CubeSats temperature. One limitation we observed from our linear state-space approach was the lack of constraints on the heating source, resulting in the designer needing to take special care when placing the pole of the system. However, we could instead employ a method like Model Predictive Control (MPC). MPC minimizes a cost function to determine the next input to the system [43]. This cost function can constrain the input resulting in the designer not having to worry about unrealistic inputs as they do with linear state-space control.

Appendix A

MATLAB Code

MATLAB code: State Space Thermal Control of CubeSats

URL: https://github.com/mklupar/State-Space-Thermal-Control-of-CubeSats

8 Node A,B,C Matrices:

A =

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	0.9348	0.00432	0.03886	2.454e-06	0.02197	2.454e-06	2.454e-06	2.454e-06	1
x2	0.00432	0.9348	2.454e-06	0.03886	2.454e-06	0.02197	2.454e-06	2.454e-06	1
x3	0.03886	2.454e-06	0.9348	0.00432	2.454e-06	2.454e-06	0.02197	2.454e-06	1
x4	2.454e-06	0.03886	0.00432	0.9348	2.454e-06	2.454e-06	2.454e-06	0.02197	1
x5	0.02197	2.454e-06	2.454e-06	2.454e-06	0.9348	0.00432	0.03886	2.454e-06	1
x6	2.454e-06	0.02197	2.454e-06	2.454e-06	0.00432	0.9348	2.454e-06	0.03886	1
x7	2.454e-06	2.454e-06	0.02197	2.454e-06	0.03886	2.454e-06	0.9348	0.00432	1
x8	2.454e-06	2.454e-06	2.454e-06	0.02197	2.454e-06	0.03886	0.00432	0.9348	1
x9	0	0	0	0	0	0	0	0	1

B =

	u1	u2	u3	u4	u5	u6	u7	u8
x1	0.0008218	0	0	0	0	0	0	0
x2	0	0.0008218	0	0	0	0	0	0
x3	0	0	0.0008218	0	0	0	0	0
x4	0	0	0	0.0008218	0	0	0	0
x5	0	0	0	0	0.0008218	0	0	0
x6	0	0	0	0	0	0.0008218	0	0
x7	0	0	0	0	0	0	0.0008218	0
x8	0	0	0	0	0	0	0	0.0008218
x9	0	0	0	0	0	0	0	0

C =

	x1	x2	x3	x4	x5	x6	x7	x8	x9
y1	1	0	0	0	0	0	0	0	0
y2	0	1	0	0	0	0	0	0	0
y3	0	0	1	0	0	0	0	0	0
y4	0	0	0	1	0	0	0	0	0
y5	0	0	0	0	1	0	0	0	0
y6	0	0	0	0	0	1	0	0	0
y7	0	0	0	0	0	0	1	0	0
y8	0	0	0	0	0	0	0	1	0
y9	0	0	0	0	0	0	0	0	1

Figure A.1: Open Loop dynamic triplet matrices A,B, and C for the eight input eight output case. The ninth state is used to model the constant term after linearizing the nonlinear dynamics.

Bibliography

[1] NASA. *State-of-the-Art of Small Spacecraft Technology*. Ames Research Center. Moffett Field, CA, 94035-1000, 2024, pp. 201–222. URL: https://www.nasa.gov/wp-content/uploads/2024/03/soa-2023.pdf?emrc=8ad1a1.

[2] A. Anvari, F. Farhani., and K.S. Niaki. "Comparative Study on Space Qualified Paints Used for Thermal Control of a Small Satellite". In: *Iranian Journal of Chemical Engineering* (2009).

[3] T.E. Brooks and H.P. Stahl. "Precision Thermal Control Technology to Enable Thermally Stable Telescopes". In: *JATIS* (2022). DOI: 10.1117/1.JATIS.8.2.024001.

[4] John E. Krist et al. "End-to-end coronagraphic modeling including a low-order wavefront sensor". In: *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*. Ed. by Mark C. Clampin et al. Vol. 8442. International Society for Optics and Photonics. SPIE, 2012, p. 844253. DOI: 10.1117/12.927143. URL: https://doi.org/10.1117/12.927143.

[5] Thomas E. Brooks and H. Philip Stahl. "Extreme dimensional stability thermal control test". In: *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts X*. Ed. by Allison A. Barto, James B. Breckinridge, and H. Philip Stahl.

Vol. 11819. International Society for Optics and Photonics. SPIE, 2021, 118190B. DOI: 10.1117/12.2593969. URL: https://doi.org/10.1117/12.2593969.

[6] Thomas E. Brooks. "Predictive thermal control applied to HabEx". In: *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*. Ed. by Howard A. MacEwen and James B. Breckinridge. Vol. 10398. International Society for Optics and Photonics. SPIE, 2017, p. 1039814. DOI: 10.1117/12.2274338. URL: https://doi.org/10.1117/12.2274338.

[7] Carlton L Foster et al. "Solar-array-induced disturbance of the Hubble space telescope pointing system". In: *Journal of Spacecraft and Rockets* 32.4 (1995), pp. 634–644.

[8] Earl A. Thornton and Yool A. Kim. "Thermally induced bending vibrations of a flexible rolled-up solar array". In: *Journal of Spacecraft and Rockets* 30.4 (1993), pp. 438–448. DOI: 10.2514/3.25550. eprint: https://doi.org/10.2514/3.25550. URL: https://doi.org/10.2514/3.25550.

[9] Steven L Brunton and J Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2022.

[10] R.P. Borase et al. "A review of PID control, tuning methods and applications". In: *International Journal of Dynamics and Control* (2020). DOI: 10.1007/s40435-020-00665-4.

[11] S. Kozák. "From PID to MPC: Control Engineering Methods Development and Applications". In: *2016 Cybernetics & Informatics (K&I)*. 2016, pp. 1–7. DOI: 10.1109/CYBERI.2016.7438634.

[12] Jamshed Iqbal et al. "Nonlinear control systems-A brief overview of historical and recent advances". In: *Nonlinear Engineering* (2017), pp. 301–312. DOI: 10.1515/nleng-2016-0077.

[13] M. S. Fadali and A. Visioli. *Digital Control Engineering: Analysis and Design*. 2nd. Elsevier, 2013.

[14] S.K. Mitter. "Filtering and stochastic control: a historical perspective". In: *IEEE Control Systems Magazine* 16.3 (1996), pp. 67–76. DOI: 10.1109/37.506400.

[15] Evgenya L Shkolnik. "On the Verge of an Astronomy CubeSat Revolution". In: *Nature Astronomy* 2.5 (2018), pp. 374–378.

[16] Michael A. Swartwout. "A brief history of rideshares (and attack of the CubeSats)". In: *2011 Aerospace Conference*. 2011, pp. 1–15. DOI: 10.1109/AERO.2011.5747233.

[17] Ewan S Douglas et al. "Design of the deformable mirror demonstration CubeSat (DeMi)". In: *Techniques and Instrumentation for Detection of Exoplanets VIII*. Vol. 10400. SPIE. 2017, pp. 285–296.

[18] Arash Mehrparvar et al. "Cubesat design specification rev. 13". In: *The CubeSat Program, Cal Poly San Luis Obispo, US* 1.2 (2014).

[19] Thyrso Villela et al. "Towards the thousandth CubeSat: A statistical overview". In: *International Journal of Aerospace Engineering* 2019.1 (2019), p. 5063145.

[20] Michael Swartwout. "You say "Picosat", I say "'CubeSat": Developing a better taxonomy for secondary spacecraft". In: *2018 IEEE Aerospace Conference*. 2018, pp. 1–17. DOI: 10.1109/AERO.2018.8396755.

[21] Lee D. Feinberg et al. "Space telescope design considerations". In: *Optical Engineering* 51.1 (2012), p. 011006. DOI: 10.1117/1.OE.51.1.011006. URL: https://doi.org/10.1117/1.OE.51.1.011006.

[22] Raymie Fotherby et al. "Thermal Performance and Design of the SPRITE Far-UV CubeSat". In: *UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XXII*. Vol. 11821. SPIE. 2021, pp. 380–390. DOI: 10.1117/12.2593760.

[23] Víctor Villalba, Hans Kuiper, and Eberhard Gill. "Review on Thermal and Mechanical Challenges in the Development of Deployable Space Optics". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 6.1 (2020), pp. 010902–010902. DOI: 10.1117/1.JATIS.6.1.010902.

[24] Jaren N Ashcraft et al. "The Versatile CubeSat Telescope: Going to Large Apertures in Small Spacecraft". In: *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts X*. Vol. 11819. SPIE. 2021, pp. 15–25.

[25] Paula do Vale Pereira et al. "Thermomechanical Design and Testing of the Deformable Mirror Demonstration Mission (DeMi) CubeSat". In: *34th Annual Small Satellite Conference* (2020).

[26] Vishal Jagota, Aman Preet Singh Sethi, and Khushmeet Kumar. "Finite element method: an overview". In: *Walailak Journal of Science and Technology (WJST)* 10.1 (2013), pp. 1–8.

[27] Ray W Clough. "Original formulation of the finite element method". In: *Finite elements in analysis and design* 7.2 (1990), pp. 89–101.

[28] J Tinsley Oden. "Historical comments on finite elements". In: *A history of scientific computing*. 1990, pp. 152–166.

[29] M Ben Salah et al. "Control volume finite element method for radiation". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 92.1 (2005), pp. 9–30.

[30] Daniel R Rousse. "Numerical predictions of two-dimensional conduction, convection, and radiation heat transfer. I. Formulation". In: *International journal of thermal sciences* 39.3 (2000), pp. 315–331.

[31] KK Gupta and JL Meek. "A brief history of the beginning of the finite element method". In: *International journal for numerical methods in engineering* 39.22 (1996), pp. 3761–3774.

[32] Ted Belytschko, Yun Yun Lu, and Lei Gu. "Element-free Galerkin methods". In: *International journal for numerical methods in engineering* 37.2 (1994), pp. 229–256.

[33] Alexander H-D Cheng and Daisy T Cheng. "Heritage and early history of the boundary element method". In: *Engineering analysis with boundary elements* 29.3 (2005), pp. 268–302.

[34] John T Katsikadelis. *The boundary element method for engineers and scientists: theory and applications*. Academic Press, 2016.

[35] George G O'Brien, Morton A Hyman, and Sidney Kaplan. "A study of the numerical solution of partial differential equations". In: *Journal of Mathematics and Physics* 29.1-4 (1950), pp. 223–251.

[36] Donald W Peaceman. "A personal retrospection of reservoir simulation". In: *A history of scientific computing*. 1990, pp. 106–129.

[37] John Crank and Phyllis Nicolson. "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type". In: *Advances in Computational Mathematics* 6.1 (1996), pp. 207–226.

[38] W.F. Ames. "Chapter 4: Numerical Methods". In: *Nonlinear Partial Differential Equations in Engineering*. Vol. 18. Mathematics in Science and Engineering. Elsevier, 1965, pp. 223–290. DOI: https://doi.org/10.1016/S0076-5392(09)60485-9. URL: https://www.sciencedirect.com/science/article/pii/S0076539209604859.

[39] Tom Moir. "Discrete-Time State-Space". In: *Feedback*. Cham: Springer International Publishing, 2020, pp. 277–302. ISBN: 978-3-030-34839-7. DOI: 10.1007/978-3-030-34839-7_11.

[40] T. L. Bergman et al. *Introduction to Heat Transfer*. 6th. Wiley, 2011.

[41] J. R. Howell, M.R. Mengüç, and R. Siegel. *Thermal Radiative Heat Transfer*. 6th. CRC Press, 2016.

[42] Thermal Desktop. *Version: 23.2*. Thermal Desktop, Inc., 2023. URL: https://www.thermaldesktop.com.

[43] Max Schwenzer et al. "Review on model predictive control: An engineering perspective". In: *The International Journal of Advanced Manufacturing Technology* 117.5 (2021), pp. 1327–1349.