Spring 2007

State Feedback Control of a Single-Loop Thermosyphon System Via a Quotient Controller

Jonathan S. Tanner

STATE FEEDBACK CONTROL OF

A SINGLE-LOOP THERMOSYPHON SYSTEM

VIA A QUOTIENT CONTROLLER

by

JONATHAN S. TANNER

(Under the Direction of Yan Wu)

ABSTRACT

The objective of this work is to design a quotient controller to stabilize a chaotic flow in a single loop thermosyphon system with a high heat index. The thermosyphon loop is heated from below and cooled from above, which causes time-dependent chaotic flow when the external heat index is above a threshold value. Our goal was to stabilize the fluid into a convective uni-directional flow well into the chaotic regime. We also investigated adding a tracking integrator to the thermosyphon system to allow "tracking" to a specific temperature.

INDEX WORDS: Chaotic Flow, Control Systems, Convective Flow, Fourier Series, Galerkin Method, Local Stability, Lorenz Equations, Quotient Controller, Runge-Kutta Method, Single-Loop Thermosyphon, State Feedback Control

STATE FEEDBACK CONTROL OF
A SINGLE-LOOP THERMOSYPHON SYSTEM
VIA A QUOTIENT CONTROLLER

by

JONATHAN S. TANNER

B.S., Georgia Institute of Technology, 2004

B.S., Georgia Southern University, 2005

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

2007

STATE FEEDBACK CONTROL OF

A SINGLE-LOOP THERMOSYPHON SYSTEM

VIA A QUOTIENT CONTROLLER

by

JONATHAN S. TANNER

Major Professor: Yan Wu

Committee: Xiezhang Li
Viktor Maymeskul
Xiao-jun Wang

Electronic Version Approved:
May 2007

ACKNOWLEDGEMENTS

This thesis has been developed over several months, and I am proud to say that this thesis is some of the best work that I have ever done. This work would not have been possible without the considerable help of several people, whose assistance was greatly appreciated.

In particular, I would like to thank Dr. Yan Wu, my advisor. Without his guidance and help, none of this would have been possible. From the hours spent explaining the background concepts needed to complete this work to the time spent helping me to prepare for the thesis defense, his help has been central.

I would also like to thank the members of my thesis committee. The time they spent looking through my thesis and helping me to fine tune the overall work has been a huge help. It would be impossible for this thesis to have the degree of accuracy that it does without their help.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Thermosyphon

A single-loop thermosyphon system is basically a torus where the distance from the center of the tube to the center of the torus is much greater than the radius of the tube. The torus is then filled with an incompressible fluid. The lower half of the loop is heated, and the upper half of the loop is cooled. Since the lower half of the loop is heated and the upper half of the loop is cooled, the fluid within the loop has a steady convective flow if the heat that is applied to the lower half of the loop is within a specific range. Because this single-loop thermosyphon system has no internal moving parts, it can be used as a very efficient means of cooling.

The main problem with using a single-loop thermosyphon as a means of cooling is that there is only a steady convective flow within the loop when the heat is below a certain level. If the loop is heated above a certain level, the system becomes chaotic and the thermosyphon is no longer an efficient means of cooling.

Lorenz Equations

To understand the behavior of the fluid flow within the loop, we examine the Navier-Stokes equations (1-3). The Navier-Stokes equations are the set of equations from fluid mechanics that govern and describe fluid flow. Although the Navier-Stokes equations fairly accurately describe the flow and behavior of fluid, the problem arises from trying to solve the Navier-Stokes equations directly. The Navier-Stokes equations are a set of three nonlinear partial differential equations. Since the set of Navier-Stokes equations are nonlinear partial differential equations, it is not possible to employ the usual methods for solving partial differential equations to find a general solution to the set of three equations that make up the Navier-Stokes equations. For example, a superposition of solutions cannot be used. Instead, each individual problem has to be examined on a case-by-case basis. Because the normal methods cannot be used to find a general solution to the nonlinear partial differential equations, the set of nonlinear partial differential equations that make up the Navier-Stokes equations need to be transformed into a form that is easier to solve. In this particular case, the nonlinear partial differential equations of the Navier-Stokes equations are transformed into the linear ordinary differential equations of the Lorenz equations.

The Lorenz equations were obtained through a nondimensionalization of variables, a change of variables to take into account the new non-dimensional variables. The derivation of the Lorenz equations is covered in more detail in a following section. One of the main problems when using the Lorenz equations to model a system is that the Lorenz equations are highly dependent on the initial conditions and can easily lead to a chaotic behavior.

Chaos

"A chaotic system is a nonlinear deterministic system that displays complex, noisy-like and unpredictable behavior."[6] There are basically two methods used to control chaotic systems: non-feedback control and feedback control. The method used here is feedback control which has many advantages over the non-feedback control method such as being more robust and not being as computationally demanding [6].

There are several different feedback controllers that are used to control chaotic systems. The particular feedback controller that is investigated here to control the chaotic system is a quotient controller. The quotient controller is added to the system to maintain stability for heat levels that would otherwise lead to a chaotic fluid flow within the thermosyphon loop.

Control Systems

In order to have a more thorough understanding of the material this thesis covers and to better understand the reasoning behind the method that was used to control the single-loop thermosyphon system, a brief introduction to basic control systems is necessary. Control systems are used in almost every aspect of life in today's society. For example, control systems are used in microwave ovens, navigation systems, space satellites, pollution control, and mass transit [4]. A control system is basically a collection of connected components that are used to provide a desired function.

The part of the control system that is going to be controlled is called the plant or process. The plant is affected by applied signals (inputs) and produces signals of interest (outputs). A controller is a device that is used to achieve a desired behavior from the plant. There are basically two types of control systems. One is referred to as open-loop or non-feedback control. In non-feedback control, the control inputs are not influenced by the outputs of the plant. The other basic type of control is called a closed-loop or feedback control. In feedback control, the system outputs are measured and fedback to the controller. The controller then compares the system outputs with the desired outputs and can make adjustments to have the actual output of the system be closer to the desired output. These differences can be more easily explained by Figures 1.1 and 1.2.

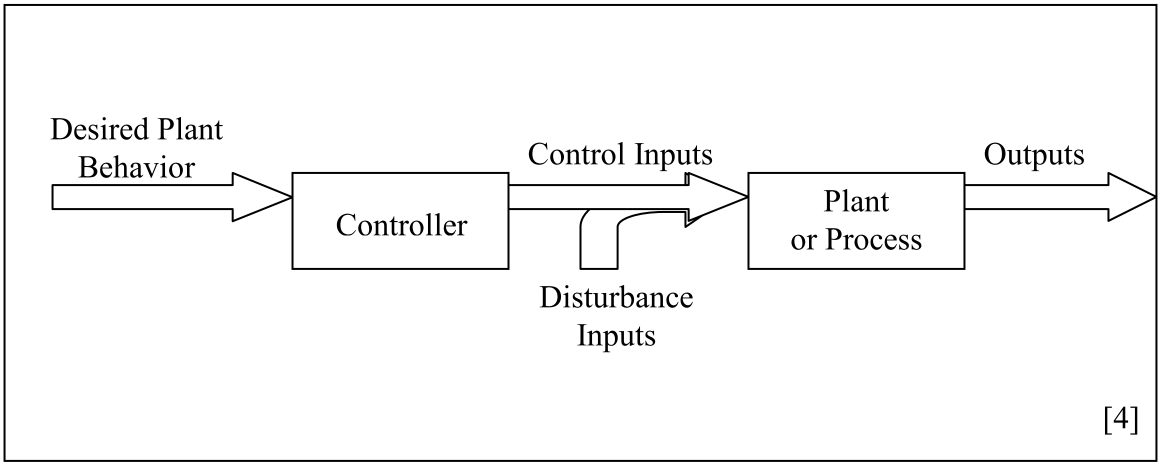

Figure 1.1 Diagram for Open-Loop or Non-Feedback Control System

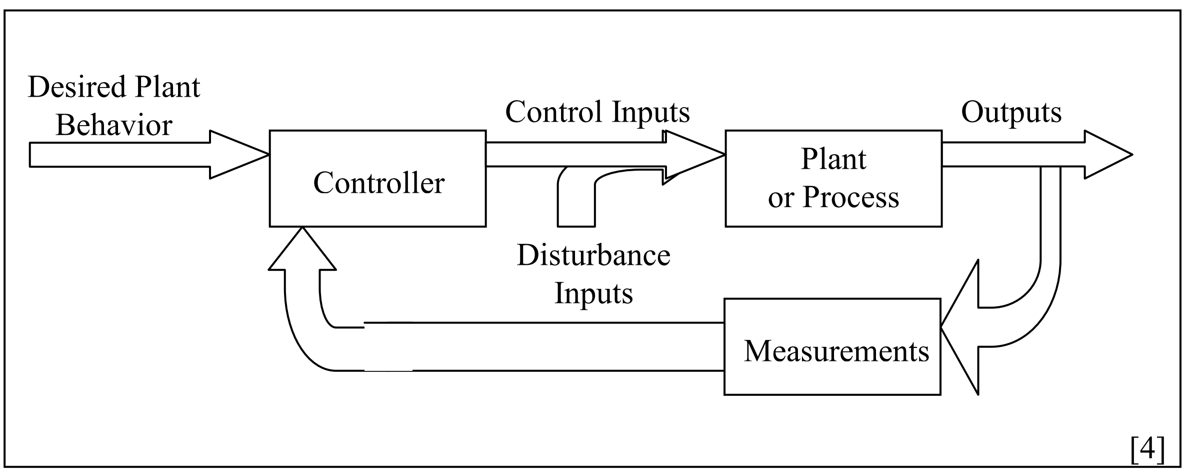

Figure 1.2 Diagram for Closed-Loop or Feedback Control System

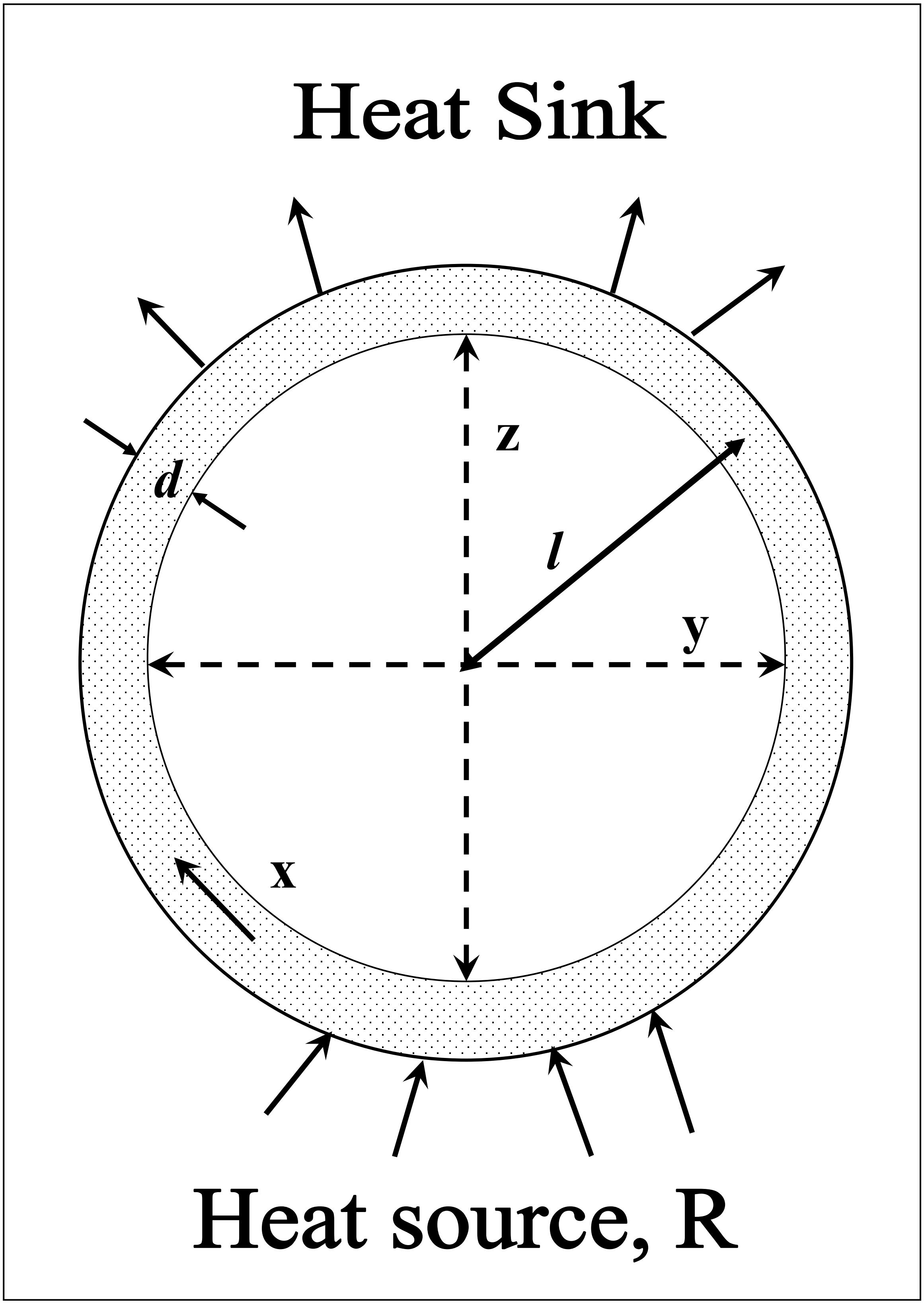

Figure 1.3. Basic Single-Loop Thermosyphon

CHAPTER 2

DERIVING LORENZ EQUATIONS

Since the diameter of the inside of the closed loop is much less than the overall diameter of the loop, d<<l, a one-dimensional modeling of the fluid flow and heat transfer within the loop is sufficient [2]. Because we are dealing with a circle and are using a one-dimensional modeling, it is convenient to use polar coordinate system to model the fluid behavior within the loop.

Starting with the Navier-Stokes equations in polar coordinates [2], the Lorenz equations that govern the single-loop thermosyphon were derived. Because in fluid mechanics θ is used to represent temperature, here ϕ is used to represent the polar angle.

$$\frac{1}{l}\frac{\partial u}{\partial \phi}=0 \tag{1}$$

$$\rho_0\left(\frac{\partial u}{\partial t}\right)=-\frac{1}{l}\frac{\partial P}{\partial \phi}-\rho(T)g\sin\phi-f_w \tag{2}$$

$$\rho_0 c_p\left(\frac{\partial T}{\partial t}+u\frac{1}{l}\frac{\partial T}{\partial \phi}\right)=h_w\left[T_w(\phi)-T\right] \tag{3}$$

From [2], we have the following equation to describe the frictional force acting on the wall of the loop.

$$f_w=\frac{\rho_0}{2}f_{w0}\,u \tag{4}$$

The first step in deriving the Lorenz equations is to nondimensionalize the variables of state and time as follows [2]:

$$t'=\frac{h_w}{\rho_0 c_p}t \tag{5}$$

$$x=\frac{\rho_0 c_p}{h_w l}u \tag{6}$$

$$y = \frac{\rho_0 c_p \alpha_0 g}{h_w\, l\, f_{w0}} S_1 \tag{7}$$

$$z = \frac{\rho_0 c_p \alpha_0 g}{h_w\, l\, f_{w0}} \left(\Delta T - C_1\right). \tag{8}$$

A Fourier series is used for the cross-sectional average temperature of the loop.

$$T(\phi,t) = T_0 + \sum_{n=1}^{\infty} \left[S_n(t)\sin(n\phi) + C_n(t)\cos(n\phi)\right] \tag{9}$$

The Galerkin method was used to simplify the Fourier series by only taking the first term in the infinite series (9).

$$T(\phi,t) = T_0 + S_1(t)\sin(\phi) + C_1(t)\cos(\phi) \tag{10}$$

A change of variables must be performed to take into account the new nondimensionalized variable of time as follows:

$$\frac{\partial u}{\partial t} = \frac{h_w}{\rho_0 c_p}\frac{\partial u}{\partial t'} \tag{11}$$

$$\frac{\partial T}{\partial t} = \frac{h_w}{\rho_0 c_p}\frac{\partial T}{\partial t'}. \tag{12}$$

The temperature distribution imposed on the loop wall, $T_w(\phi)$, is obtained from (9). Since the vertical temperature difference is always (T_H-T_C) regardless of the value of the polar angle, we can write $C_n = (T_H - T_C) = \Delta T$. The horizontal temperature difference is either (T_H-T_H) or (T_C-T_C); therefore, we can write $S_n = 0$. Finally the Galerkin method is used to take only the first term in the infinite series.

$$T_w(\phi) = T_0 + \Delta T \cos\phi \tag{13}$$

Solving (6-8) for u, C_1, and S_1, the following nondimensional variables for velocity, vertical and horizontal temperature difference are obtained.

$$u = \frac{h_w l}{\rho_0 c_p} x \tag{14}$$

$$S_1 = \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \tag{15}$$

$$C_1 = \Delta T - \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z \tag{16}$$

Starting with the left-hand-side of (3), the nondimensionalized variables (5-8) are substituted into the equation along with the equation for the cross-sectional average temperature of the loop (10). The change of variables (11&12) must also be included to take into account the new nondimensional variables.

The non-dimensional variable for u (14) is substituted into the left-hand-side of (3) and then the change of variables (11-12) is performed to obtain the following equation.

$$\rho_0 c_p \left(\frac{\partial T}{\partial t} + u \frac{1}{l} \frac{\partial T}{\partial \phi} \right) = \rho_0 c_p \left(\frac{h_w}{\rho_0 c_p} \frac{\partial T}{\partial t'} + \frac{h_w l}{\rho_0 c_p} x \frac{1}{l} \frac{\partial T}{\partial \phi} \right) = h_w \left(\frac{\partial T}{\partial t'} + x \frac{\partial T}{\partial \phi} \right)$$

Substituting the formula for cross-sectional temperature from Fourier series and Galerkin method (10) leads to the following expression.

$$h_w \left[\frac{\partial}{\partial t} \left(T_0 + S_1 \sin\phi + C_1 \cos\phi \right) + x \frac{\partial}{\partial \phi} \left(T_0 + S_1 \sin\phi + C_1 \cos\phi \right) \right]$$

Then substituting the non-dimensional values for Fourier coefficients (15-16) into the above equation yields the following expression.

$$h_w \left(\frac{\partial}{\partial t} \left(T_0 + \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \sin\phi + \Delta T \cos\phi - \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z \cos\phi \right) \right)$$
$$+ h_w x \frac{\partial}{\partial \phi} \left(T_0 + \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \sin\phi + \Delta T \cos\phi - \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z \cos\phi \right)$$

$$= h_w \left(\frac{\partial T_0}{\partial t} + \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \sin\phi \frac{\partial}{\partial t} y + \frac{\partial}{\partial t}\left(\Delta T \cos\phi\right) - \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \cos\phi \frac{\partial}{\partial t} z \right)$$

$$+ h_w \left(x \frac{\partial T_0}{\partial \phi} + x \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \frac{\partial}{\partial \phi} \sin\phi + x\Delta T \frac{\partial}{\partial \phi}\left(\cos\phi\right) - xz \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \frac{\partial}{\partial \phi} \cos\phi \right)$$

Now, cancelling terms yields the following expression.

$$h_w \left(\frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \sin\phi \frac{\partial}{\partial t} y - \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \cos\phi \frac{\partial}{\partial t} z \right)$$

$$+ h_w \left(x + x \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \frac{\partial}{\partial \phi} \sin\phi + x\Delta T \frac{\partial}{\partial \phi}\left(\cos\phi\right) - xz \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \frac{\partial}{\partial \phi} \cos\phi \right)$$

By performing differentiation, the following can be obtained.

$$\frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \dot{y} \sin\phi - \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \dot{z} \cos\phi + \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} xy \cos\phi - h_w x \Delta T \sin\phi + \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} xz \sin\phi$$

$$= \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \dot{y} \sin\phi - \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \dot{z} \cos\phi + \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} xy \cos\phi$$

$$- x\Delta T \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \left(\frac{\rho_0 c_p \alpha_0 g}{h_w\, l\, f_{w0}} \right) \sin\phi + \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} xz \sin\phi$$

By grouping sine and cosine terms, (17) is obtained.

$$\frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} \left(\dot{y} + \left(z - \frac{\rho_0 c_p \alpha_0 g}{h_w\, l\, f_{w0}} \Delta T \right) x \right) \sin\phi + \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} (xy - \dot{z}) \cos\phi \tag{17}$$

Substituting formula for cross-sectional temperature from Fourier series and Galerkin method (10) into the right-hand-side of (3) allows us to obtain the following equation.

$$h_w \left[T_w(\phi) - T \right] = h_w \left[T_w(\phi) - (T_0 + S_1 \sin\phi + C_1 \cos\phi) \right]$$

By substituting the formula for the temperature distribution imposed on the loop wall (13), we obtain the equation below.

$$h_w\left[T_0 + \Delta T\cos\phi - (T_0 + S_1\sin\phi + C_1\cos\phi)\right] = h_w\left[\Delta T\cos\phi - S_1\sin\phi - C_1\cos\phi\right]$$

Substituting the non-dimensional values for Fourier coefficients (15-16) leads to (18).

$$h_w\left[\Delta T\cos\phi - \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y\sin\phi - \left(\Delta T - \frac{h_w\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z\right)\cos\phi\right]$$

$$= h_w\Delta T\cos\phi - \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y\sin\phi - h_w\Delta T\cos\phi - \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z\cos\phi$$

$$= \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z\cos\phi - \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y\sin\phi \qquad (18)$$

Since both sides of (3) were reduced to only sines and cosines, we can set the coefficients of $\cos\phi$ equal to each other and the coefficients of $\sin\phi$ equal to each other to obtain the Lorenz equations for horizontal and vertical temperature difference.

Setting the $\cos\phi$ coefficients equal to each other yields the following equation.

$$\frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z = \frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g}\left(xy - \dot{z}\right)$$

Cancelling terms and solving for $\dot{z}$ gives (19).

$$\dot{z} = xy - z \qquad (19)$$

Setting the $\sin\phi$ coefficients equal to each other yields the following equation.

$$\frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g}\left(\dot{y} + \left(z - \frac{\rho_0 c_p \alpha_0 g}{h_w\, l\, f_{w0}}\Delta T\right)x\right) = -\frac{h_w^2\, l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y$$

Cancelling terms and solving for $\dot{y}$ gives (20).

$$\dot{y} = Rx - y - xz \quad , \quad R = \frac{\rho_0 c_p \alpha_0 g \Delta T}{h_w\, l\, f_{w0}} \tag{20}$$

Working with the momentum equation (2), substitute (4) to describe the frictional force acting on the wall of the loop and substitute non-dimensional variables into the equation.

$$\rho_0\left(\frac{h_w}{\rho_0 c_p}\frac{\partial u}{\partial t'}\right) = -\frac{1}{l}\frac{\partial P}{\partial \phi} - \rho(T) g \sin\phi - \frac{\rho_0 f_{w0}}{2}\left(\frac{h_w\, l}{\rho_0 c_p} x\right)$$

We also need to use the Boussinesq approximation in describing the fluid density, *ρ(T)*. The Boussinesq approximation assumes that all fluid properties are independent of temperature with the exception of fluid density, which varies linearly with temperature.

$$\rho(T) = \rho_0\left[1 - \alpha_0 (T - T_0)\right] \tag{21}$$

After substituting the Boussinesq approximation (21) into the momentum equation, we get

$$\rho_0\left(\frac{h_w}{\rho_0 c_p}\frac{\partial u}{\partial t'}\right) = -\frac{1}{l}\frac{\partial P}{\partial \phi} - \rho_0\left[1 - \alpha_0 (T - T_0)\right] g \sin\phi - \frac{\rho_0 f_{w0}}{2}\left(\frac{h_w\, l}{\rho_0 c_p} x\right).$$

Now substitute the formula for cross-sectional averaged temperature (10) obtained from Fourier series and Galerkin method and perform change of variables to take into account new non-dimensional variables to get the following equation.

$$\frac{h_w}{c_p}\frac{\partial}{\partial t}\left(\frac{h_w\, l}{\rho_0 c_p} x\right) = -\frac{1}{l}\frac{\partial P}{\partial \phi} - \rho_0 g\left[1 - \alpha_0 (T_0 + S_1 \sin\phi + C_1 \cos\phi - T_0)\right]\sin\phi - \frac{f_{w0} h_w\, l}{2 c_p} x$$

Then canceling terms and performing differentiation we arrive at (22).

$$\frac{h_w^2\, l}{\rho_0 c_p^2}\dot{x} = -\frac{1}{l}\frac{\partial P}{\partial \phi} - \rho_0 g \sin\phi + \rho_0 g \alpha_0 S_1 \sin^2\phi + \rho_0 g \alpha_0 C_1 \sin\phi\cos\phi - \frac{f_{w0} h_w\, l}{2 c_p} x \qquad (22)$$

Now substitute (15) and (16) for S_1 and C_1, respectively, and integrate (22) around loop to remove the φ dependence and obtain the velocity equation.

$$\int_0^{2\pi} \frac{h_w^2\, l}{\rho_0 c_p^2}\dot{x}\, d\phi = -\frac{1}{l}\int_0^{2\pi}\frac{\partial P}{\partial \phi} d\phi - \int_0^{2\pi} \rho_0 g \sin\phi\, d\phi + \int_0^{2\pi} \rho_0 g \alpha_0 \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} y \sin^2\phi\, d\phi$$

$$+\int_0^{2\pi} \rho_0 g \alpha_0\, \Delta T \sin\phi\cos\phi\, d\phi - \int_0^{2\pi} \rho_0 g \alpha_0 \frac{h_w l\, f_{w0}}{\rho_0 c_p \alpha_0 g} z \sin\phi\cos\phi\, d\phi - \int_0^{2\pi} \frac{f_{w0} h_w\, l}{2 c_p} x\, d\phi$$

Performing the integration around the loop, leads to the following equation.

$$\frac{2\pi\, h_w^2\, l}{\rho_0 c_p^2}\dot{x} = -\frac{1}{l}\left[P\right]_{\phi=0}^{\phi=2\pi} + \rho_0 g \alpha_0 \frac{h_w\, l\, f_{w0}}{\rho_0 g \alpha_0 c_p} y\pi - \frac{f_{w0} h_w\, l}{2 c_p} 2\pi\, x$$

Since φ =2 π is the same point as φ=0, we have P(2π) = P(0). Therefore P(2 π)-P(0)=0.

$$\frac{2\pi\, h_w^2\, l}{\rho_0 c_p^2}\dot{x} = \rho_0 g \alpha_0 \frac{h_w\, l\, f_{w0}}{\rho_0 g \alpha_0 c_p} y\pi - \frac{f_{w0} h_w\, l}{2 c_p} 2\pi\, x$$

Performing other cancellations we obtain (23).

$$\dot{x} = \frac{\rho_0 c_p f_{w0}}{2 h_w}(y - x) \qquad (23)$$

Rewriting (19), (20), and (23), we now have the Lorenz equations governing the single-loop thermosyphon.

$$\begin{array}{ll} \dot{x} = p(y-x), & p = \dfrac{\rho_0 c_p f_{w0}}{2 h_w} \\ \dot{y} = Rx - y - xz, & R = \dfrac{\rho_0 c_p \alpha_0 g \Delta T}{h_w\, l\, f_{w0}} \\ \dot{z} = xy - bz, & b = 1 \end{array} \qquad (24)$$

CHAPTER 3

LOCAL STABILITY ANALYSIS

Once we have the Lorenz equations, we need to perform a stability analysis of the equations to determine what is required to have a stable thermosyphon system.

Working with Lorenz equations

$$\dot{x} = p(y - x) \quad (24a)$$
$$\dot{y} = Rx - y - xz \quad (24b)$$
$$\dot{z} = xy - z \quad (24c)$$

The first main step in performing the local stability analysis is to find $\vec{x}_{ss}$, the three steady-state equilibria. We define state vector that describes the system as $\vec{x} = \langle x, y, z \rangle$ and the steady-state state vector as $\vec{x}_{ss} = \langle x_{ss}, y_{ss}, z_{ss} \rangle$.

The steady state is the constant value that the function with have at time equal to infinity.

$$\lim_{t \to \infty} \vec{x} = \vec{x}_{ss} = const. \Rightarrow \frac{d}{dt}\vec{x} = 0$$

Applying the steady state condition to (24a-c) and then solving the three equations for the three unknowns yields (25-27).

$$z_{ss} = R\text{-}1 \quad (25)$$

$$y_{ss} = \pm\sqrt{R-1} \quad (26)$$

$$x_{ss} = \pm\sqrt{R-1} \quad (27)$$

From (29-31), we can see that there are three equilibria which we define below.

$x_{ss} = 0$	$x_{ss} = \sqrt{R-1}$	$x_{ss} = -\sqrt{R-1}$
$y_{ss} = 0$	$y_{ss} = \sqrt{R-1}$	$y_{ss} = -\sqrt{R-1}$
$z_{ss} = 0$	$z_{ss} = R-1$	$z_{ss} = R-1$

The second main step in performing the local stability analysis is to examine the local stability of $\vec{x}_{ss}$, the three steady-state equilibria that were found. In order to examine the local stability, we first introduce a small time dependent perturbation vector, $\delta\vec{x} = \delta\vec{x}(t)$: $\vec{x} = \vec{x}_{ss} + \delta\vec{x}$, where $\delta\vec{x} = \langle \delta x, \delta y, \delta z \rangle$.

We now have $\vec{x} = \langle x_{ss} + \delta x, y_{ss} + \delta y, z_{ss} + \delta z \rangle$ (28)

Using (24a) and (28), the following can be obtained.

$$\dot{x} = p(y - x) \Rightarrow \frac{d}{dt}(x_{ss} + \delta x) = p(y_{ss} + \delta y - x_{ss} - \delta x)$$

From calculations, we know $x_{ss}=y_{ss}$, so we can cancel terms to obtain:

$$\delta\dot{x} = -p\delta x + p\delta y \ . \quad (29)$$

Using (24b) and (28), we obtain the following:

$$\dot{y} = Rx - y - xz \Rightarrow \frac{d}{dt}(y_{ss} + \delta y) = Rx_{ss} + R\delta x - y_{ss} - \delta y - (x_{ss} + \delta x)(z_{ss} + \delta z)$$

$$\delta\dot{y} = Rx_{ss} + R\delta x - y_{ss} - \delta y - (x_{ss}z_{ss} + z_{ss}\delta x + x_{ss}\delta z + \delta x\delta y) \ .$$

Since δx and δz are very small, δx δz is sufficiently small to be neglected.

$$\delta\dot{y} = Rx_{ss} + R\delta x - y_{ss} - \delta y - x_{ss}z_{ss} - z_{ss}\delta x - x_{ss}\delta z$$

$$\delta\dot{y} = (Rx_{ss} - y_{ss} - x_{ss}z_{ss}) - z_{ss}\delta x - x_{ss}\delta z + R\delta x - \delta y$$

From calculations, we know ($Rx_{ss}-y_{ss}-x_{ss}z_{ss}=0$), so we can cancel terms to obtain (30).

$$\delta\dot{y} = (R - z_{ss})\delta x - \delta y - x_{ss}\delta z \quad (30)$$

Using (24c) and (32), yields the following:

$$\dot{z} = xy - z \Rightarrow \frac{d}{dt}(z_{ss} + \delta z) = (x_{ss} + \delta x)(y_{ss} + \delta y) - (z_{ss} + \delta z)$$

$$\delta\dot{z} = (x_{ss}y_{ss} - z_{ss}) + y_{ss}\delta x + x_{ss}\delta y + \delta x\delta y - \delta z \ .$$

From calculations, we know ($x_{ss}y_{ss}$-z_{ss}=0) and x_{ss}=y_{ss}, so we can obtain the following equation.

$$\delta\dot{z} = x_{ss}\delta x + x_{ss}\delta y + \delta x\delta y - \delta z$$

Since δx and δy are very small, δx δy is sufficiently small to be neglected.

$$\delta\dot{z} = x_{ss}\delta x + x_{ss}\delta y - \delta z \tag{31}$$

From (29-31) we can get $\delta\dot{\vec{x}} = J\delta\vec{x}$, or in matrix form:

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{y} \\ \delta\dot{z} \end{bmatrix} = \begin{bmatrix} -p & p & 0 \\ (R - z_{ss}) & -1 & -x_{ss} \\ x_{ss} & x_{ss} & -1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix}$$

where the Jacobian matrix, $J = \begin{bmatrix} -p & p & 0 \\ (R - z_{ss}) & -1 & -x_{ss} \\ x_{ss} & x_{ss} & -1 \end{bmatrix}$. (32)

Since solutions to the differential equations will have the form

$x = e^{\lambda_1 t}v_1 + e^{\lambda_2 t}v_2 + ... + e^{\lambda_n t}v_n$, we need to have Re{$\lambda_i$}≤0 so that $x \neq \infty$ as $t \to \infty$.

We need to examine individual cases for each equilibrium point.

<u>case(i)</u>: $x_{ss} = y_{ss} = z_{ss} = 0$
<u>case(ii)</u>: $x_{ss} = y_{ss} = \sqrt{R-1}$ $z_{ss} = R-1$
<u>case(iii)</u>: $x_{ss} = y_{ss} = -\sqrt{R-1}$ $z_{ss} = R-1$

Using the standard convention, we set p=10 for the Prandtl number and using the steady state values for case(i), (32) can be rewritten as:

$$J = \begin{bmatrix} -10 & 10 & 0 \\ R & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Then we find the eigenvalues of J.

$$|J - \lambda I| = \begin{vmatrix} -10-\lambda & 10 & 0 \\ R & -1-\lambda & 0 \\ 0 & 0 & -1-\lambda \end{vmatrix} = 0$$

$$(-10-\lambda)(-1-\lambda)(-1-\lambda) - 10(R(-1-\lambda)) = 0$$

$$(-1-\lambda)(\lambda^2 + 11\lambda + 10(1-R)) = 0 \qquad (33)$$

From (33), we can find the first eigenvalue: λ_1= -1. To find the other two eigenvalues, we need to use the quadratic formula and (33). Using the quadratic formula to find the roots of $\lambda^2+11\lambda+10(1\text{-}R) = 0$, we arrive at the following eigenvalues.

$$\lambda_2 = \frac{-11 + \sqrt{81+40R}}{2} \qquad\qquad \lambda_3 = \frac{-11 - \sqrt{81+40R}}{2}$$

λ_3 will always have a negative value since the Rayleigh number, R, is always positive, so λ_2 will be the limiting eigenvalue because it may be positive if $\sqrt{81+40R}$ is sufficiently large.

Now that we know the λ_2 is the limiting eigenvalue, we need to solve for the conditions to make λ_2 lie in the left-hand-side of the complex plane. Once we know the conditions necessary to keep λ_2 in the left-hand-side of the complex plane, we will have the conditions necessary to keep the system stable under case (i).

$$\lambda_2 = \frac{-11 + \sqrt{81+40R}}{2} \le 0 \Rightarrow \sqrt{81+40R} \le 11 \Rightarrow 81 + 40R \le 121 \Rightarrow R \le 1 \qquad (34)$$

From (34), we see that for case (i) the system is unstable when R>1.

$$\boxed{x_{ss} = y_{ss} = z_{ss} = 0 \text{ system is unstable for } R>1} \tag{35}$$

Using the standard convention, we set p=10 for the Prandtl number and the steady state values for case (ii), (32) can be rewritten as:

$$J = \begin{bmatrix} -10 & 10 & 0 \\ 1 & -1 & -\sqrt{R-1} \\ \sqrt{R-1} & \sqrt{R-1} & -1 \end{bmatrix}.$$

Then we find the eigenvalues of J.

$$|J - \lambda I| = \begin{vmatrix} -10-\lambda & 10 & 0 \\ 1 & -1-\lambda & -\sqrt{R-1} \\ \sqrt{R-1} & \sqrt{R-1} & -1-\lambda \end{vmatrix} = 0 \tag{36}$$

After taking the determinate, (36) yields the following characteristic polynomial:

$$\lambda^3 + 12\lambda^2 + (R+10)\lambda + (20R-20) = 0. \tag{37}$$

Using the standard convention, we again set p=10 for the Prandtl number then the steady state values for case (iii), (32) can be rewritten as:

$$J = \begin{bmatrix} -10 & 10 & 0 \\ 1 & -1 & \sqrt{R-1} \\ -\sqrt{R-1} & -\sqrt{R-1} & -1 \end{bmatrix}.$$

Then we find the eigenvalues of J.

$$|J - \lambda I| = \begin{vmatrix} -10-\lambda & 10 & 0 \\ 1 & -1-\lambda & \sqrt{R-1} \\ -\sqrt{R-1} & -\sqrt{R-1} & -1-\lambda \end{vmatrix} = 0 \tag{38}$$

After taking the determinate, (38) yields the same characteristic polynomial, (37), that we obtained from case (ii). Since case (ii) and case (iii) have the same characteristic polynomial for finding the eigenvalues, solving (37) for values of Re{λ}≤0 will give us the conditions required to make the system stable for case (ii) and case (iii).

Because (37) is a cubic polynomial that cannot be easily factored or solved directly, we used Routh-Hurwitz Testing [4] to determine what values of R will give values of Re{λ}≤0. Routh-Hurwitz testing is a method that can be used to determine how many eigenvalues of a given system lie on the right-hand-side of the complex plane.

For Routh-Hurwitz testing, we construct a table using the coefficients of the characteristic polynomial and then look for sign changes in the first column of the table. The number of sign changes in the first column will be equal to the number of eigenvalues that lie on the right-hand-side of the complex plane. Since no sign changes in the first column will mean that all of the eigenvalues for the system lie either in the left-hand-side of the complex plane or on the imaginary axis, we can solve for values of R that will cause us to have no sign changes in the first column of our table in order to determine the conditions necessary to have the system be at least marginally stable.

The Routh-Hurwitz table for (37) will have the following form:

λ^3	1	(10+R)	0
λ^2	12	(20R-20)	0
λ^1	a_1	a_2	
λ^0	a_3		

where $a_1 = \dfrac{-\begin{vmatrix} 1 & (10+R) \\ 12 & (20R-20) \end{vmatrix}}{12}$, $a_2 = \dfrac{-\begin{vmatrix} (10+R) & 0 \\ (20R-20) & 0 \end{vmatrix}}{12}$, $a_3 = \dfrac{-\begin{vmatrix} 12 & (20R-20) \\ a_1 & a_2 \end{vmatrix}}{a_1}$.

By taking the determinate and substituting the values for a_1 and a_2, we get the following values for a_1, a_2 and a_3.

$$a_1 = \frac{(35 - 2R)}{3} \qquad a_2 = 0 \qquad a_3 = 20R - 20 \qquad \text{(39a-c)}$$

Substituting (39a-c), the Routh-Hurwitz table above can be rewritten as:

λ^3	1	(10+R)	0
λ^2	12	(20R-20)	0
λ^1	(35-2R)/3	0	
λ^0	(20R-20)		

In order to have the system be at least marginally stable, we need to have no sign changes in the first column of the Routh-Hurwitz table. This means that there are two conditions that need to be met to have a system that is a least marginally stable.

$$(35 - 2R)/3 \geq 0 \qquad \text{and} \qquad (20R - 20) \geq 0$$

Solving for R, the conditions for stability can be rewritten as

(i) $R \leq 35/2$ (ii) $R \geq 1$.

Therefore, by combining (i) and (ii), we arrive at the following condition to have a thermosyphon system that is at least marginally stable.

$$\boxed{1 \leq R \leq 35/2}$$

CHAPTER 4

SIMULATION

After determining what values of the Rayleigh number are needed to have stability in the Thermosyphon system, we needed to simulate the system in order to verify the results from the stability analysis. The single-loop thermosyphon system was simulated in MATLAB using the Fourth -Order Runge-Kutta method to solve the ordinary differential equation. The Fourth -Order Runge-Kutta method is a numerical method for solving ordinary differential equations, and is given by the formulae below.

$$x(t+h)=x(t)+\frac{1}{6}\left(F_1+2F_2+2F_3+F_4\right),\ where$$

$$\begin{aligned} F_1 &= hf(t,x) \\ F_2 &= hf(t+\tfrac{1}{2}h, x+\tfrac{1}{2}F_1) \\ F_3 &= hf(t+\tfrac{1}{2}h, x+\tfrac{1}{2}F_2) \\ F_4 &= hf(t+h, x+F_3) \end{aligned} \quad [5] \tag{40}$$

The numerical method that we used to solve the ordinary differential equations is called a fourth-order method because it reproduces the terms in Taylor's series up to and including the h^4 term. Because the Fourth -Order Runge-Kutta method cannot take into account all of the terms in Taylor's series, there is a truncation error inherent in this method of numerical integration. Since the Fourth-Order Runge-Kutta method does not take into account the terms in Taylor's series beyond the h^4 term, the truncation error for the Fourth -Order Runge-Kutta method is of $O(h^5)$ [5].

The first program that was created simulated the behavior of the single-loop thermosyphon system with no controller. Once the MATLAB code for the single-loop thermosyphon was written, we were ready to simulate the behavior of the system to see if the stability analysis was correct. To achieve this, we created a set of three bifurcation plots. We had a bifurcation plot for the velocity of the fluid within the loop, x, the horizontal temperature difference, y, and the vertical temperature difference, z. The plots showed the behavior of the system over different values of the Rayleigh number, R. An example of the bifurcation plots can be seen in Figure 4.1.

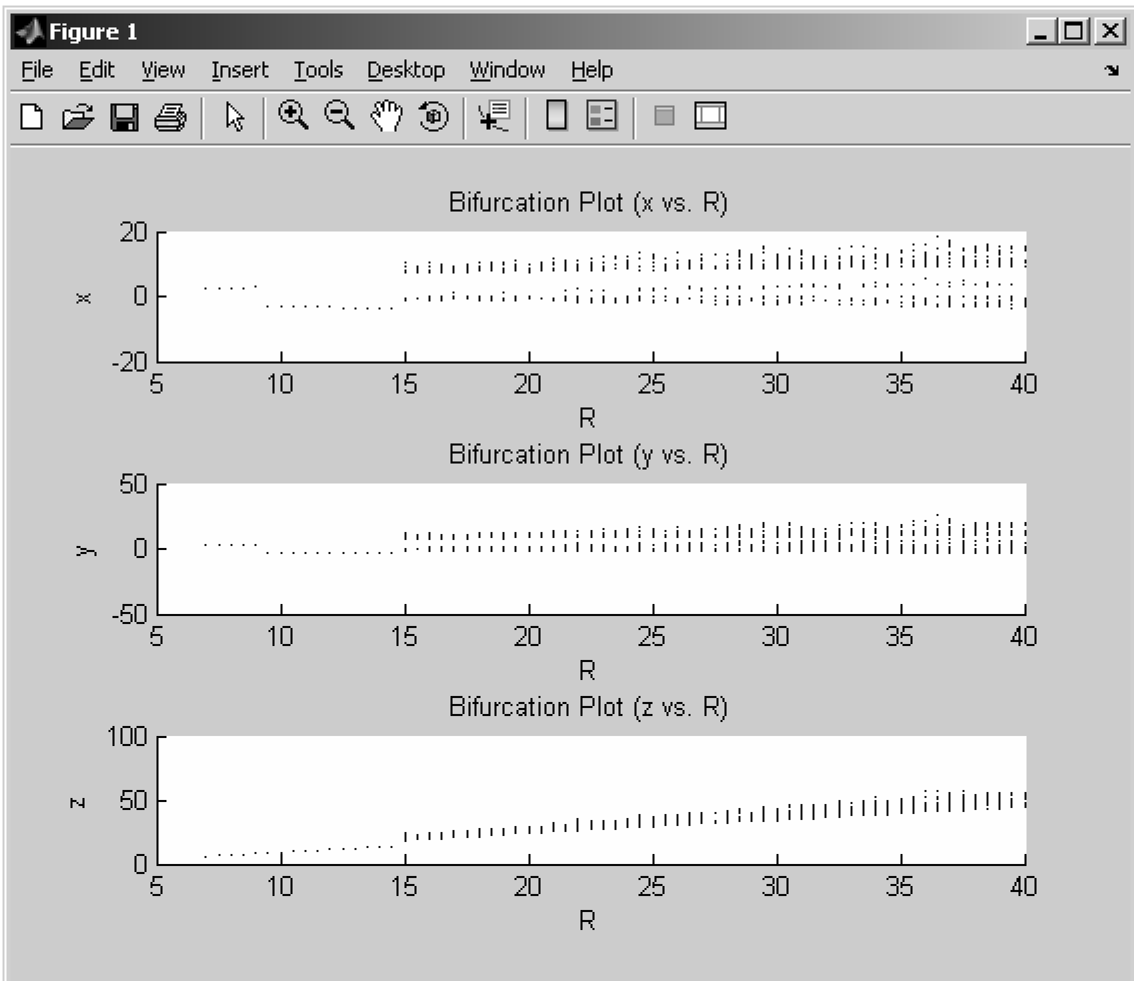

Figure 4.1. Bifurcation Plots for x, y, z vs. R

Since the Rayleigh number refers to the heat index imposed at the bottom of the loop, we can see that the system is stable if the heat that is applied to the bottom of the loop is below a given value. More specifically, the bifurcation plot shows that the thermosyphon system is stable for values of $R<15$. This simulation verifies that the fluid flow within the thermosyphon becomes chaotic for values of R that are greater than a certain value. However, the simulation differs slightly from what we found analytically. The simulation shows that the system is chaotic for $R\geq15$, but analytically we found that the system would be stable for $R\leq35/2=17.5$. The slight difference between what we found analytically and what the simulation shows can be explained by the strong dependence of the Lorenz equations on the initial conditions, the inherent error present in the numerical methods used in the simulation and by the limited precision of MATLAB. The complete MATLAB code for the all of the simulations can be seen in the Appendix.

From the bifurcation plot, we can see the behavior of the thermosyphon system for many different values of the Rayleigh number. However, to attain a more thorough understanding of the thermosyphon system, we need to observe the behavior of the system as it changes over time. In order to achieve this, we created a set of plots that shows how the velocity, horizontal temperature difference, and vertical temperature difference change over time for a specific value of the Rayleigh number. In Figures 4.2-4.5, we can see the behavior of the system for $R=10$.

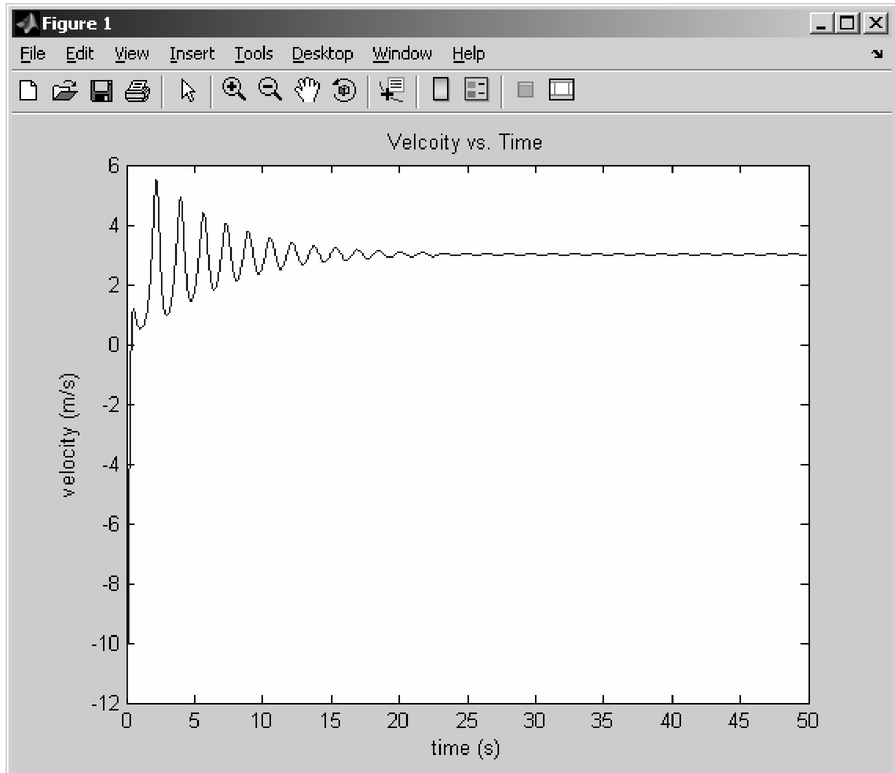

Figure 4.2 Plot for System Behavior x vs. time with R=10

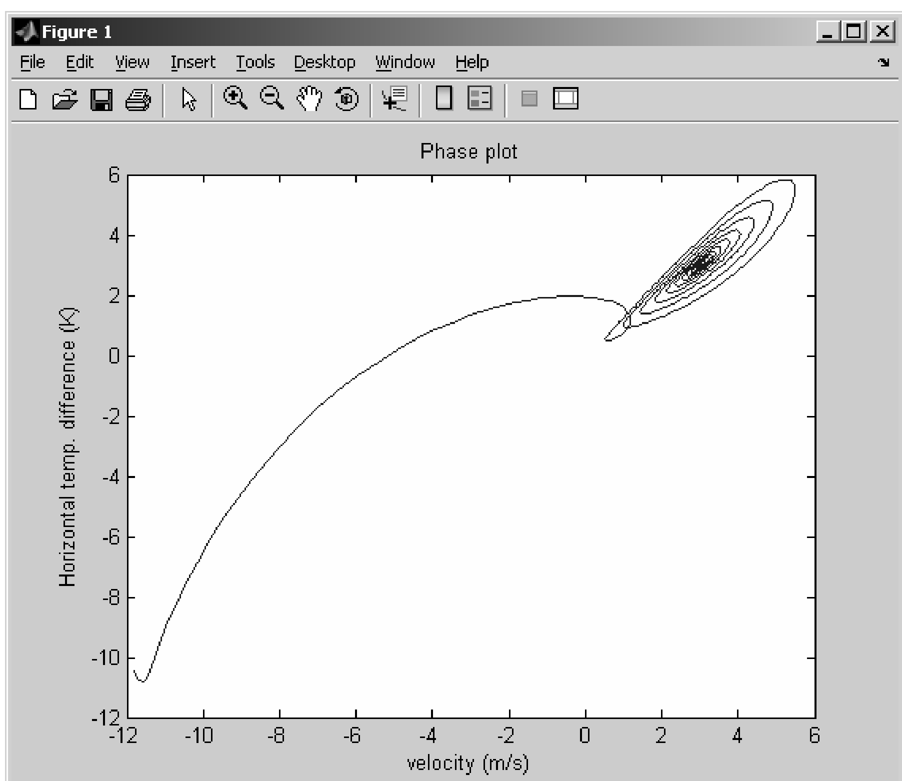

Figure 4.3 Phase Plot for System Behavior x vs. y with R=10

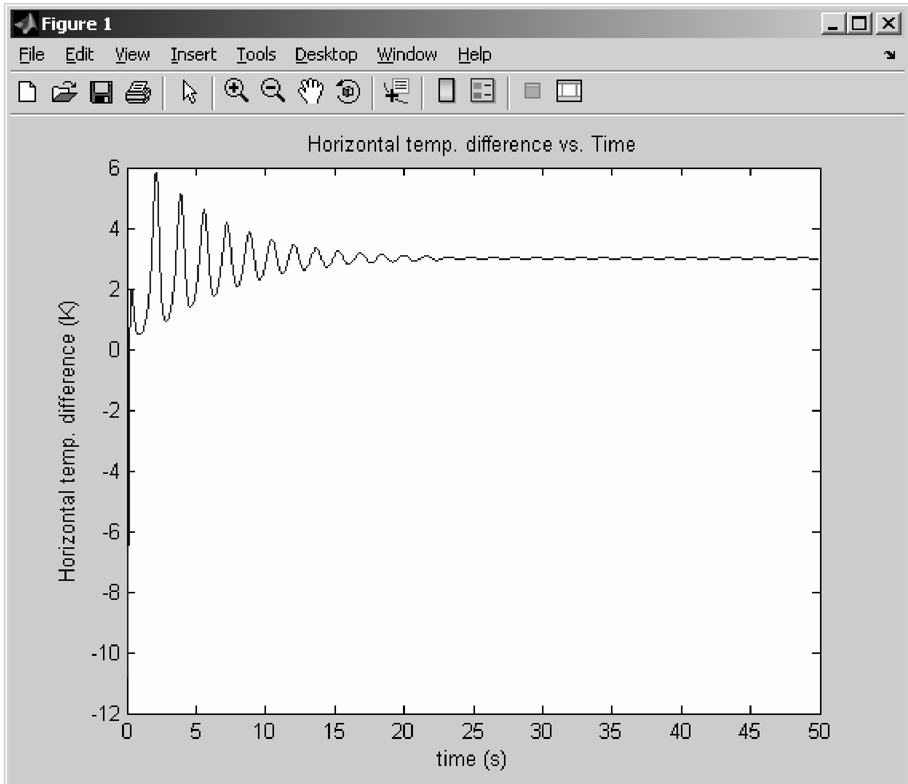

Figure 4.4 Plot for System Behavior y vs. time with R=10

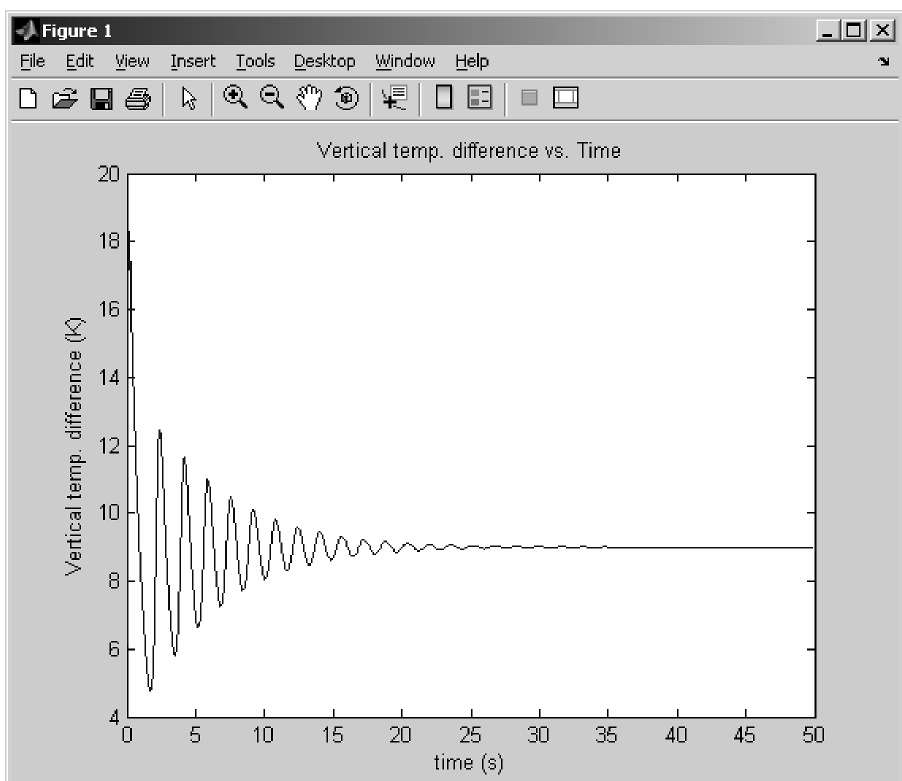

Figure 4.5 Plot for System Behavior z vs. time with R=10

The plots in Figures 4.2-4.5 were obtained by giving the system some arbitrary initial conditions and setting R = 10. The value of R that was used is well within the required range for the system to be stable, and we can easily see from Figures 4.2-4.5 that initially the system has an oscillatory behavior that quickly decays to one of the steady-state equilibria. In this particular case the steady-state equilibrium was $x_{ss} = y_{ss} = \sqrt{\mathrm{R} - 1}\;\; z_{ss} = R - 1$.

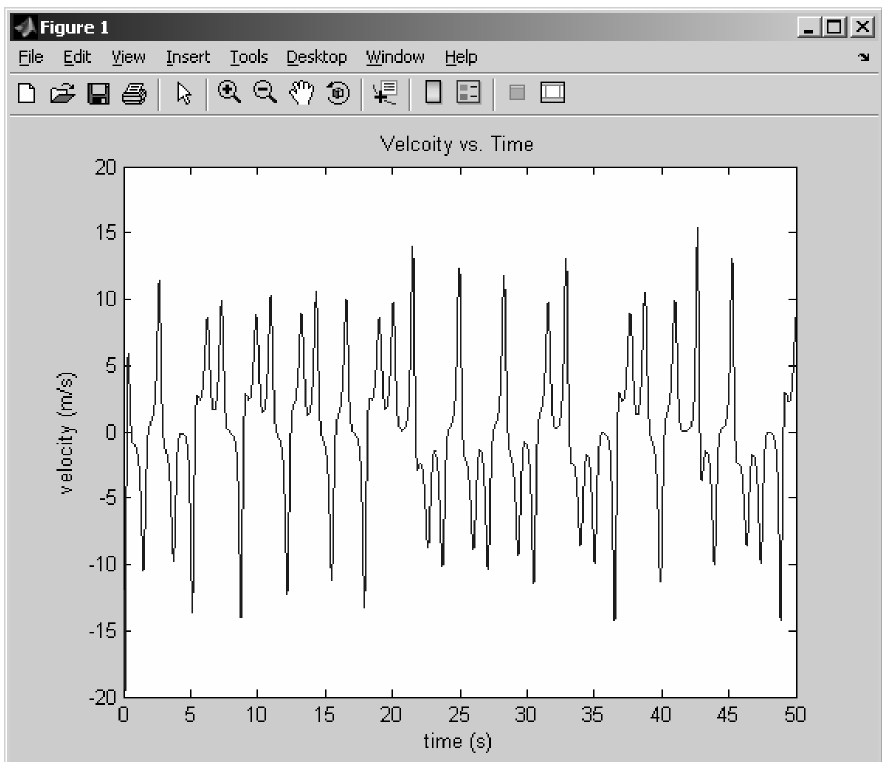

Figure 4.6 Plot for System Behavior x vs. time with R=30

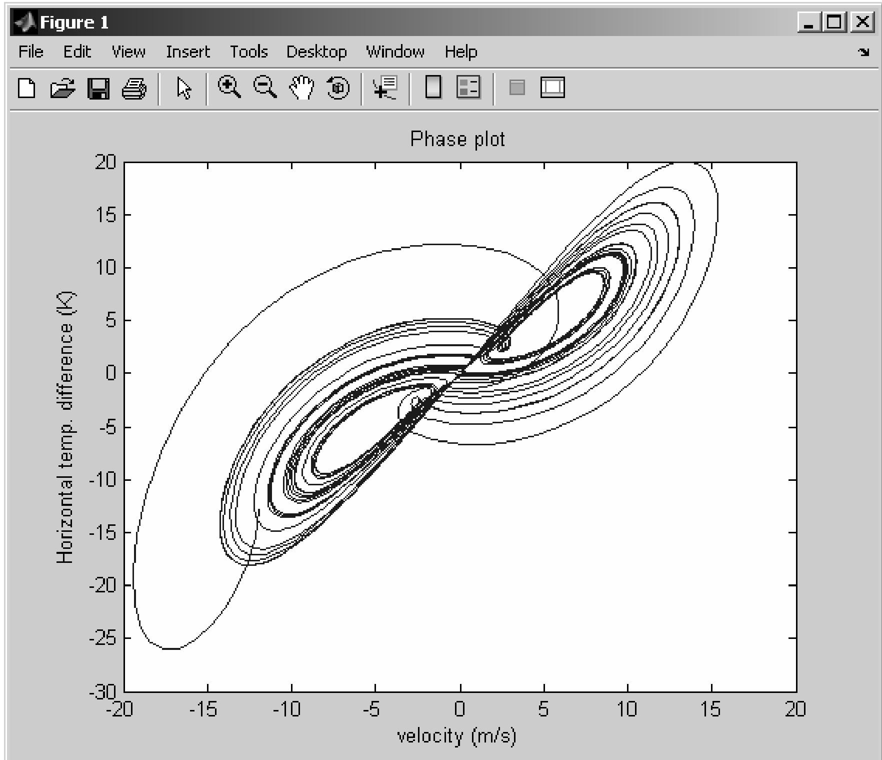

Figure 4.7 Phase Plot for System Behavior x vs. y with R=30

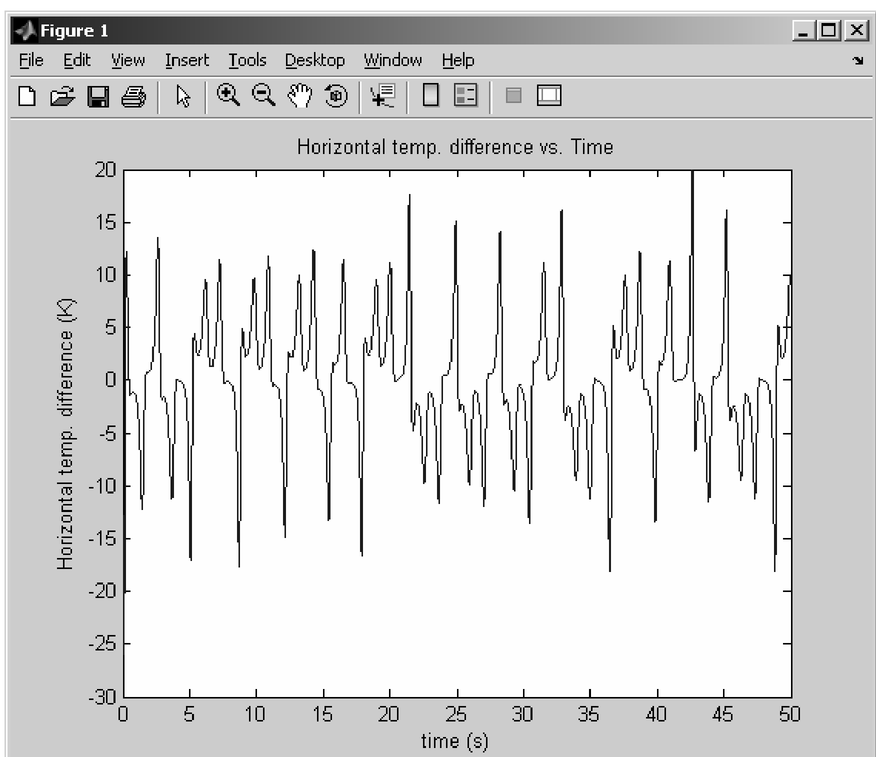

Figure 4.8 Plot for System Behavior y vs. time with R=30

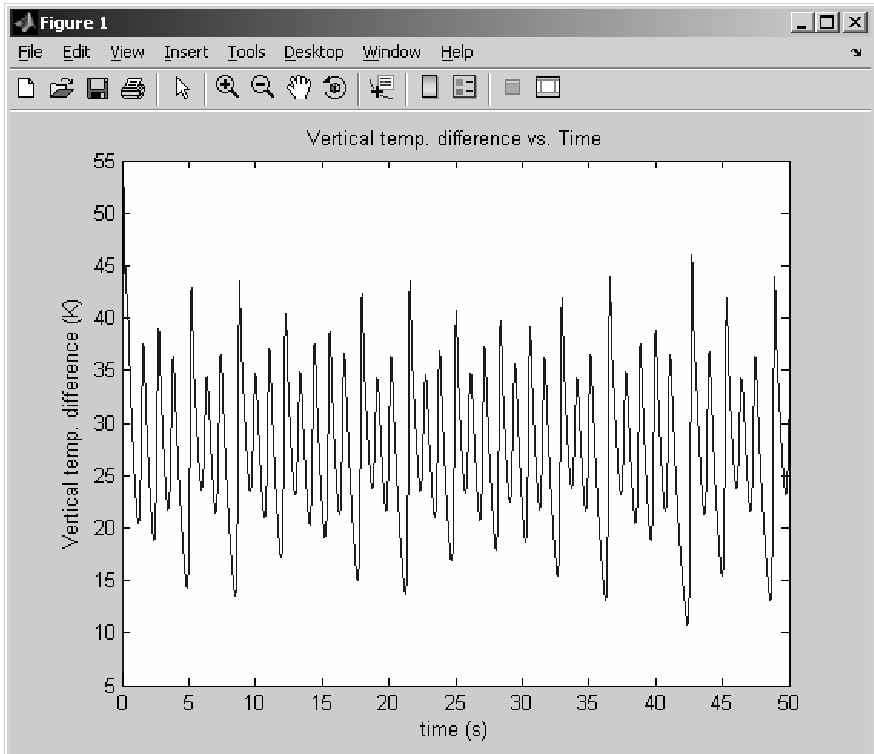

Figure 4.9 Plot for System Behavior z vs. time with R=30

In Figures 4.6-4.9, we can see the behavior of the system for R=30. Again the system was given arbitrary initial conditions, but this time the value of R was set equal to 30. The value of R that was used for the Figures 4.6-4.9 plots was well above the region that would make the system behave in a chaotic manner, and as we can see from Figures 4.6-4.9, the system begins with an oscillatory behavior and then quickly becomes more chaotic.

CHAPTER 5

QUOTIENT CONTROLLER

The controller that is used to return stability to the thermosyphon system once the fluid flow in the thermosyphon becomes chaotic is a quotient controller with the form seen in (41).

$$u = -k\left(\frac{y^2}{z}\right) \tag{41}$$

In the equation for the quotient controller, y describes the horizontal temperature difference in the thermosyphon loop, z describes the vertical temperature difference and k is the gain of the quotient controller. The vertical temperature difference in the loop is always a positive value related to (T_h-T_c), where T_h is the temperature in the lower half of the loop and T_c is the temperature in the upper half of the loop. Since T_h will always be a higher temperature than T_c, the quantity (T_h-T_c) is always positive. The vertical temperature difference is usually a positive number with the exception mentioned above where T_h=T_c. In this case, we will have z=0 which is the trivial steady-state solution for the thermosyphon system and does not require a controller to return stability to the system.

For the quotient controller to return stability to the system, it is added to the value of R, in the Lorenz Equations that govern fluid flow within the thermosyphon. R is the Rayleigh number which refers to the heat index imposed at the bottom of the thermosyphon loop, so adding the quotient controller to R is like a small perturbation to the amount of heat in the system. Because the controller has the form seen in (41), where y and z are representative of temperature differences of the loop and k is the gain of the controller, which only depends on R or the heat index of the system, y, z, and k can all be easily determined from measurements of the thermosyphon system. Therefore, the value of the quotient controller can be easily and completely determined from temperatures measurements taken from the system. After adding the quotient controller, the Lorenz equations can now be rewritten as follows:

$$\dot{x} = p(y-x) \tag{42a}$$
$$\dot{y} = (R+u)x - y - xz \tag{42b}$$
$$\dot{z} = xy - z\,. \tag{42c}$$

Now that we have the quotient controller included in the Lorenz equations that govern the fluid flow inside the thermosyphon loop, we need to perform local stability analysis on these new Lorenz equations (42) the same way that the stability analysis was performed on the Lorenz equations that did not contain the quotient controller. The first step in performing the local stability analysis is to find $\vec{x}_{ss} = \langle x_{ss}, y_{ss}, z_{ss} \rangle$, the three steady-state equilibria. The steady state is the constant value that the function with have at time equal to infinity.

$$\lim_{t\to\infty} \vec{x} = \vec{x}_{ss} = const. \Rightarrow \frac{d}{dt}\vec{x} = 0$$

Applying the steady state conditions to (42a-c), and then solving the three equations for the three unknowns we get (43-45).

$$z_{ss} = R - k - 1 \tag{43}$$

$$y_{ss} = \pm\sqrt{R-k-1} \tag{44}$$

$$x_{ss} = \pm\sqrt{R-k-1} \tag{45}$$

From (43-45), we can see that there are three equilibria which we define below.

$x_{ss}=0$ $y_{ss}=0$ $z_{ss}=0$	$x_{ss}=\sqrt{R-k-1}$ $y_{ss}=\sqrt{R-k-1}$ $z_{ss}=R-k-1$	$x_{ss}=-\sqrt{R-k-1}$ $y_{ss}=-\sqrt{R-k-1}$ $z_{ss}=R-k-1$

The second main step in performing the local stability analysis is to examine the local stability of $\vec{x}_{ss}$, the three steady-state equilibria that were found. Once the quotient controller is added, the system becomes too complicated to use the perturbation approach that was used previously to determine the local stability of $\vec{x}_{ss}$. Therefore to determine the local stability of the system with the quotient controller, we used the Jacobian approach where the Jacobian is given by (46).

$$J(x,y,z)=\begin{bmatrix} \frac{\partial\dot{x}}{\partial x} & \frac{\partial\dot{x}}{\partial y} & \frac{\partial\dot{x}}{\partial z} \\ \frac{\partial\dot{y}}{\partial x} & \frac{\partial\dot{y}}{\partial y} & \frac{\partial\dot{y}}{\partial z} \\ \frac{\partial\dot{z}}{\partial x} & \frac{\partial\dot{z}}{\partial y} & \frac{\partial\dot{z}}{\partial z} \end{bmatrix} \tag{46}$$

Using (42) and (46) and simplifying, we rewrite the Jacobian matrix as:

$$J=\begin{bmatrix} -p & p & 0 \\ 1 & -(2k+1) & \frac{k}{x_{ss}}-x_{ss} \\ x_{ss} & x_{ss} & -1 \end{bmatrix}. \tag{47}$$

Since solutions to the differential equations will have the form $x=e^{\lambda_1 t}v_1+e^{\lambda_2 t}v_2+...+e^{\lambda_n t}v_n$, we need to have $\mathrm{Re}\{\lambda_i\}<0$ so that $x\neq\infty$ as $t\to\infty$.

We need to examine individual cases for each equilibrium point.

<u>case(i)</u>: $x_{ss} = y_{ss} = z_{ss} = 0$
<u>case(ii)</u>: $x_{ss} = y_{ss} = \sqrt{\mathrm{R - k - 1}}\,,\ z_{ss} = R - k - 1$
<u>case(iii)</u>: $x_{ss} = y_{ss} = -\sqrt{\mathrm{R - k - 1}},\ z_{ss} = R - k - 1$

Using the standard convention we set p=10 for the Prandtl number. Then rewriting (47) we find the characteristic polynomial as follows:

$$|J - \lambda I| = \begin{vmatrix} -10-\lambda & 10 & 0 \\ 1 & -(2k-1)-\lambda & \left(\dfrac{k}{x} - x\right) \\ x & x & -1-\lambda \end{vmatrix} = 0$$

$$\lambda^3 + (12+2k)\lambda^2 + (11+21k+x_{ss}^2)\lambda + 20x_{ss}^2 = 0\,. \qquad (48)$$

Since case (ii) and case (iii) have the same values for x^2_{ss}, we can examine the local stability of these two equilibria at the same time. Substituting the value for x^2_{ss} from case (ii) and case (iii) into (48), the characteristic polynomial becomes:

$$\lambda^3 + (12+2k)\lambda^2 + (10+20k+R)\lambda + 20(R-k-1) = 0\,. \qquad (49)$$

From the characteristic polynomial, we need to determine what values for the gain of the quotient controller, k, will give eigenvalues in the left-hand-side of the complex plane, so the system will be stable. The restrictions on the gain of the quotient controller are found by using Routh-Hurwitz Testing.

For Routh-Hurwitz testing, we construct a table using the coefficients of the characteristic polynomial and then look for sign changes in the first column of the table. The number of sign changes in the first column will be equal to the number of eigenvalues that lie on the right-hand-side of the complex plane. Since no sign changes in the first column will mean that all of the eigenvalues for the system lie either in the left-hand-side of the complex plane or on the imaginary axis, we can solve for values of k that will cause us to have no sign changes in the first column of our table in order to determine the conditions necessary to have the system be at least marginally stable.

The Routh-Hurwitz table for (49) will have the following form:

λ^3	1	(10+20k+R)	0
λ^2	(12+2k)	(20R-20k-20)	0
λ^1	a_1	a_2	
λ^0	a_3		

where $a_1 = \dfrac{-\begin{vmatrix} 1 & (10+20k+R) \\ (12+2k) & (20R-20k-20) \end{vmatrix}}{(12+2k)}$, $a_2 = \dfrac{-\begin{vmatrix} (10+20k+R) & 0 \\ (20R-20K-20) & 0 \end{vmatrix}}{(12+2k)}$,

$$a_3 = \frac{-\begin{vmatrix} (12+2k) & (20R-20k-20) \\ a_1 & a_2 \end{vmatrix}}{a_1}.$$

By taking the determinate and substituting the values for a_1 and a_2, we get the following values for a_1, a_2 and a_3.

$$a_1 = \frac{40k^2 + (280+2R)k + (140-8R)}{(12+2k)} \qquad a_2 = 0 \qquad a_3 = 20(R-k-1) \qquad \text{(50a-c)}$$

Substituting (50a-c), the Routh-Hurwitz table above can be rewritten as:

λ^3	1	(10+20k+R)	0
λ^2	(12+2k)	20(R-k-1)	0
λ^1	$[40k^2+(280+2R)k+(140-8R)]/(12+2k)$	0	
λ^0	20(R-k-1)		.

In order to have the system be marginally stable, we need to have no sign changes in the first column of the Routh-Hurwitz table. This means that there are three conditions that need to be met to have a system that is a least marginally stable.

$$(12+2k) \geq 0\,, \qquad \frac{40k^2+(280+2R)k+(140-8R)}{(12+2k)} \geq 0\ , \qquad 20(R-k-1) \geq 0$$

Since the gain for the quotient controller is always a non-negative number, the first condition will be satisfied by any value of k. This leaves us with two constraints that will be used to determine what values we need for the gain of the quotient controller to return stability to our system.

Solving for k, the conditions for stability can be rewritten as follows:

$$\text{(i)}\ \ k \geq \frac{-280-2R+\sqrt{4R^2+2400R+56000}}{80} \qquad \text{(ii)} \quad k \leq R-1\,.$$

Therefore, by combining (i) and (ii), we can arrive at the following condition required to have a thermosyphon system with the quotient controller that is at least marginally stable.

$$\boxed{\frac{-280-2R+\sqrt{4R^2+2400R+56000}}{80} \leq k \leq R-1}$$

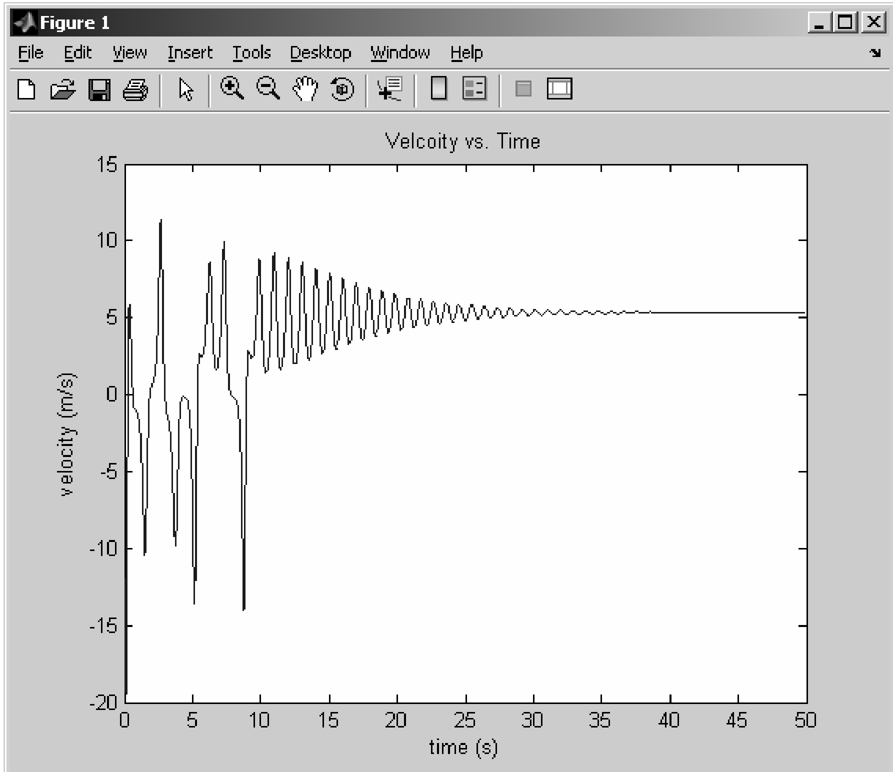

Figure 5.1 Plot of x vs. time with Quotient Controller and R=30

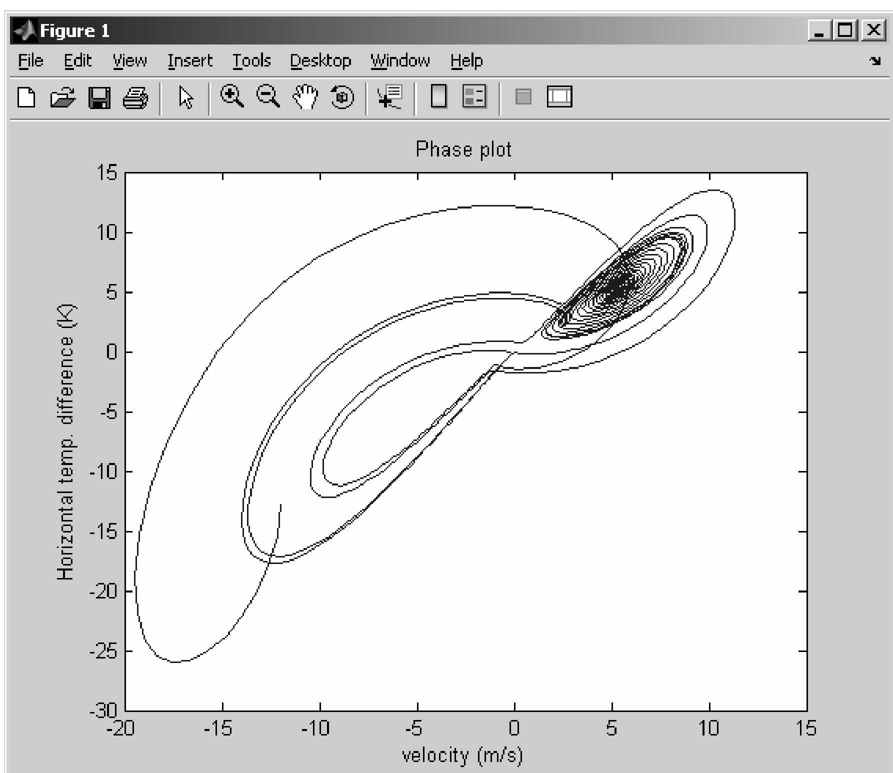

Figure 5.2 Plot of x vs. y with Quotient Controller and R=30

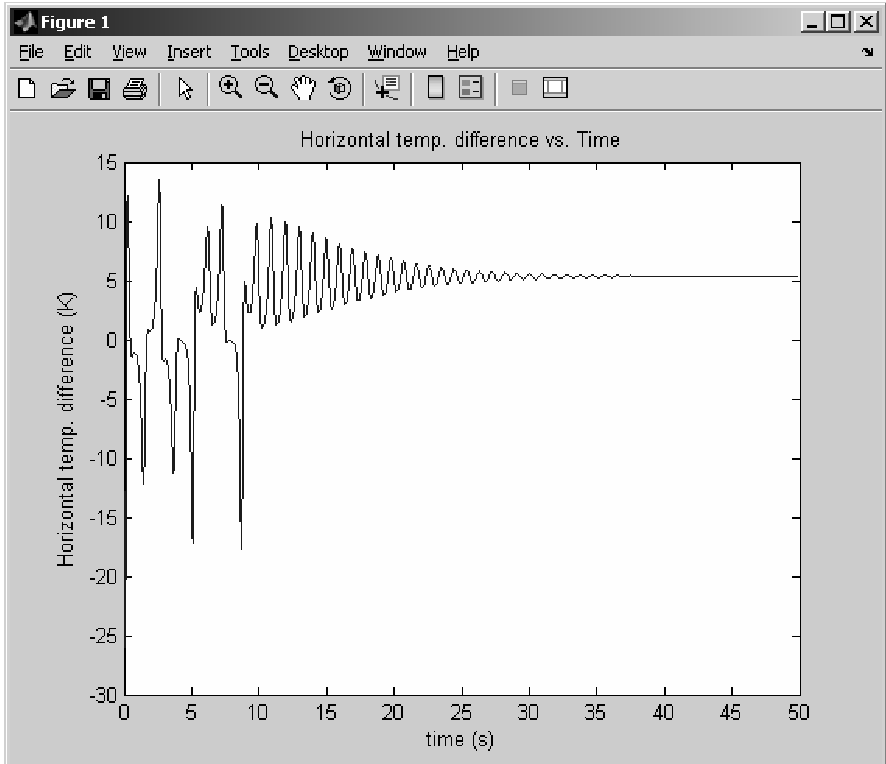

Figure 5.3 Plot of y vs. time with Quotient Controller and R=30

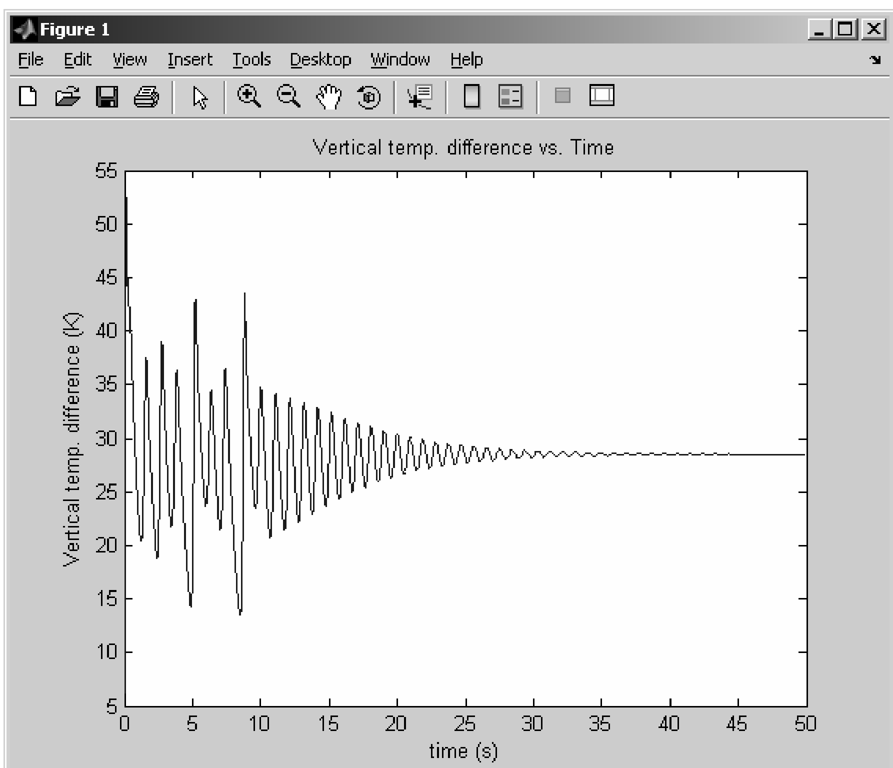

Figure 5.4 Plot of z vs. time with Quotient Controller and R=30

After determining the values for the gain of the quotient controller needed to keep the thermosyphon stable, it was necessary to verify the results with a simulation in MATLAB. Figures 4.6-4.9 show that with no controller the thermosyphon system becomes chaotic when R=30. To demonstrate that the quotient controller can return stability to a chaotic system, the simulation was run with R=30 and no controller and then after a set number of iterations the controller was added to the system. From Figures 5.1-5.4, it is easy to see that the system is becoming more chaotic until the controller is activated (approximately at t=10s) and the oscillatory behavior begins to decay until the system reaches one of the steady-state equilibria. In this particular case, the steady-state equilibrium was $x_{ss} = y_{ss} = \sqrt{\mathrm{R - k - 1}}\ z_{ss} = R - k - 1$.

CHAPTER 6

TRACKING

The tracking integrator is added to the thermosyphon system to allow us to "track" to a specific state for the system. If a quotient controller with tracking is used to control the single-loop thermosyphon system, stability would be able to be returned to the chaotic system and we would have the added advantage of being able to track to a specific state. For example, we could have a stable system and then tune the controller so that the velocity of the fluid within the thermosyphon loop is within a specific range. Once the tracking is added to the system, the stable system that was achieved using the quotient controller is destabilized and a proportional controller has to be added to the system in order to return stability. The tracking integrator, t_r, has the form seen in (51),

$$t_r = -dw \tag{51}$$

where d is the gain of the integrator and w can be found from (52).

$$\dot{w} = y - y_r \tag{52}$$

The value y_r in (52) is the value that is being tracked to, and y is the state variable that corresponds to the horizontal temperature difference of the system.

The proportional controller, p_c, is defined by (53).

$$p_c = -l(y - y_r) \tag{53}$$

In the equation for the proportional controller, l is the gain for the controller, y_r is the value that is being tracked to, and y is the state variable that corresponds to the horizontal temperature difference of the system. For the tracking integrator to track to a specific value and still maintain the stability of the system, the tracking integrator along with the quotient and proportional controllers are added to the value of R in the Lorenz Equations that govern fluid flow within the thermosyphon. R is the Rayleigh number which refers to the heat index imposed at the bottom of the thermosyphon loop, so adding the tracking integrator and the quotient and proportional controllers to R is like a small perturbation to the amount of heat in the system. After adding the tracking integrator and the proportional controller to R, the Lorenz system of equations that govern the fluid flow within the single-loop thermosyphon can be rewritten as follows:

$$\dot{x} = p(y - x) \tag{54a}$$

$$\dot{y} = \left(R - k\frac{y^2}{z} - dw - l(y - y_r) \right) x - y - xz \tag{54b}$$

$$\dot{z} = xy - z \tag{54c}$$

$$\dot{w} = y - y_r \, . \tag{54d}$$

Now that we have the tracking included in the Lorenz equations that govern the fluid flow inside the thermosyphon loop, we need to perform local stability analysis on this new system of equations (54) the same way that the stability analysis was performed on the Lorenz equations that did not contain the tracking. The first step in performing the local stability analysis is to find $\vec{x}_{ss} = \langle x_{ss}, y_{ss}, z_{ss}, w_{ss} \rangle$, the steady-state equilibrium. The steady state is the constant value that the function with have at time equal to infinity.

$$\lim_{t \to \infty} \vec{x} = \vec{x}_{ss} = const. \Rightarrow \frac{d}{dt}\vec{x} = 0$$

Applying the steady state condition to (54a-d), and then solving the four equations for the four unknowns, we get (56-59).

$$y_{ss} = y_r \tag{56}$$

$$w_{ss} = \frac{R - k - 1 - y_r^2}{d} \tag{57}$$

$$x_{ss} = y_r \tag{58}$$

$$z_{ss} = y_r^2 \tag{59}$$

Grouping (56-59) we see that we have only one equilibrium.

$$\boxed{\begin{aligned} x_{ss} &= y_r \\ y_{ss} &= y_r \\ z_{ss} &= y_r^2 \\ w_{ss} &= \frac{R-k-1-y_r^2}{d} \end{aligned}}$$

The second main step in performing the local stability analysis is to examine the local stability of $\vec{x}_{ss}$, the steady-state equilibrium that was found. Once the quotient controller and tracking are added, the system becomes too complicated to use the perturbation approach to determine the local stability of $\vec{x}_{ss}$, so to determine the local stability of the system with the quotient controller and tracking, we used the Jacobian approach where the Jacobian is given by (60).

$$J(x,y,z)=\begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial w} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} & \frac{\partial \dot{y}}{\partial w} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial x} \\ \frac{\partial \dot{w}}{\partial x} & \frac{\partial \dot{w}}{\partial y} & \frac{\partial \dot{w}}{\partial z} & \frac{\partial \dot{w}}{\partial w} \end{bmatrix} \tag{60}$$

$$J(x,y,z)=\begin{bmatrix} -p & p & 0 & 0 \\ 1 & (-2k-ly_r-1) & \left(\frac{k}{y_r}-y_r\right) & -dy_r \\ y_r & y_r & -1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{61}$$

Using the standard convention we set p=10 for the Prandtl number. Then rewriting (61) we find the characteristic polynomial as follows:

$$|J(x,y,z)-I\lambda| = \begin{bmatrix} -10-\lambda & 10 & 0 & 0 \\ 1 & (-2k-ly_r-1)-\lambda & \left(\dfrac{k}{y_r}-y_r\right) & -dy_r \\ y_r & y_r & -1-\lambda & 0 \\ 0 & 1 & 0 & 0-\lambda \end{bmatrix} = 0$$

$$\lambda^4 + (2k+ly_r+12)\lambda^3 + (21k+dy_r+11ly_r+11+y_r^2)\lambda^2 + (10ly_r+11dy_r+20y_r^2)\lambda + 10dy_r = 0 \qquad (62)$$

From the characteristic polynomial (62), we need to determine what values for the gain of the integrator, d, and the gain of the proportional controller, l, will give eigenvalues in the left-hand-side of the complex plane, so the system will be stable. The restrictions on the gain of the quotient controller are found by using Routh-Hurwitz Testing.

For Routh-Hurwitz testing, we construct a table using the coefficients of the characteristic polynomial and then look for sign changes in the first column of the table. The number of sign changes in the first column will be equal to the number of eigenvalues that lie on the right-hand-side of the complex plane. Since no sign changes in the first column will mean that all of the eigenvalues for the system lie either in the left-hand-side of the complex plane or on the imaginary axis, we can solve for values of k that will cause us to have no sign changes in the first column of our table in order to determine the conditions necessary to have the system be at least marginally stable.

The Routh-Hurwitz table for (62) will have the following form:

λ^4	1	$(21k+dy_r+11ly_r+11+y_r^2)$	$10dy_r$
λ^3	$(2k+ly_r+12)$	$(10ly_r+11dy_r+20yr2)$	0
λ^2	a_1	a_2	a_3
λ^1	a_4	a_5	
λ^0	a_6		

where $$a_1 = \frac{-\begin{vmatrix} 1 & (21k + dy_r + 11ly_r + 11 + y_r^2) \\ (2k + ly_r + 12) & (10ly_r + 11dy_r + 20y_r^2) \end{vmatrix}}{(2k + ly_r + 12)},$$

$$a_2 = \frac{-\begin{vmatrix} (21k + dy_r + 11ly_r + 11 + y_r^2) & 10dy_r \\ (10ly_r + 11dy_r + 20y_r^2) & 0 \end{vmatrix}}{(2k + ly_r + 12)}, \quad a_3 = \frac{-\begin{vmatrix} 10dy_r & 0 \\ 0 & 0 \end{vmatrix}}{(2k + ly_r + 12)},$$

$$a_4 = \frac{-\begin{vmatrix} (2k + ly_r + 12) & (10ly_r + 11dy_r + 20y_r^2) \\ a_1 & a_2 \end{vmatrix}}{a_1}, \quad a_5 = \frac{-\begin{vmatrix} (10ly_r + 11dy_r + 20y_r^2) & 0 \\ a_2 & a_3 \end{vmatrix}}{a_1},$$

$$a_6 = \frac{-\begin{vmatrix} a_1 & a_2 \\ a_4 & a_5 \end{vmatrix}}{a_4}.$$

By taking the determinate and substituting the values for a_1 and a_2, we get the following values for a_1, a_2, a_3, a_4, a_5 and a_6.

$$a_1 = \frac{1}{2k + ly_r + 12}\left(133ly_r + dy_r - 8y_r^2 + 42k^2 + 43kly_r + 274k + 2dy_rk + dy_r^2l + 11l^2y_r^2 + 132 + 2y_r^2k + y_r^3l\right)$$

$$a_2 = a_6 = \frac{10dy_r\left(10ly_r + 20y_r^2\right)}{2k + ly_r + 12}$$

$$a_3 = a_5 = 0$$

$$a_4 = \frac{-y_r(10l + 11d + 20y_r)\left(-18dy_rk - 9dy_r^2l - 119dy_r + 133ly_r - 8y_r^2 + 42k^2 + 43kly_r + 274k + 11l^2y_r^2 + 132 + 2y_r^2k + y_r^3l\right)}{-133ly_r - dy_r + 8y_r^2 - 42k^2 - 43kly_r - 274k - 2dy_rk - dy_r^2l - 11l^2y_r^2 - 132 - 2y_r^2k - y_r^3l}$$

In order to have the system be marginally stable, we need to have no sign changes in the first column of the Routh-Hurwitz table. This means that there are four conditions that need to be met to have a system that is a least marginally stable.

(i) $(2k+ly_r+12) \geq 0$ (ii) $a1 \geq 0$ (iii) $a_4 \geq 0$ (iv) $a_6 \geq 0$

Since all of the conditions for local stability (i-iv) have at least two values for the gain, it can easily be shown that there are an infinite number of combinations for the gain of the quotient controller, tracking integrator and proportional controller that will lead to a system that is at least marginally stable.

After examining the local stability of the Lorenz system with the quotient controller, tracking integrator, and the proportional controller, it was necessary to verify the results with a simulation in MATLAB. Figures 4.6-4.9 show that with no controller the thermosyphon system becomes chaotic when R=30. To demonstrate that the quotient controller with tracking can return stability to a chaotic system and track to a specific value, the simulation was run with R=45 and no controller. After a set number of iterations, the controller was added to the system. Once stability was returned to the system, some time was allowed to pass, and the tracking was turned on. From Figures 6.1-6.3, it is easy to see that the system is becoming more chaotic until the controller is activated (approximately at t=40s) and the oscillatory behavior begins to decay until the system reaches the steady-state equilibrium. Then the tracking integrator is turned on (approximately at t=60s). It is easy to see from Figures 6.1-6.3 that the system is initially perturbed, but then quickly goes to the value that is being tracked to. In this particular case the value that is being tracked to is y_r=13.54.

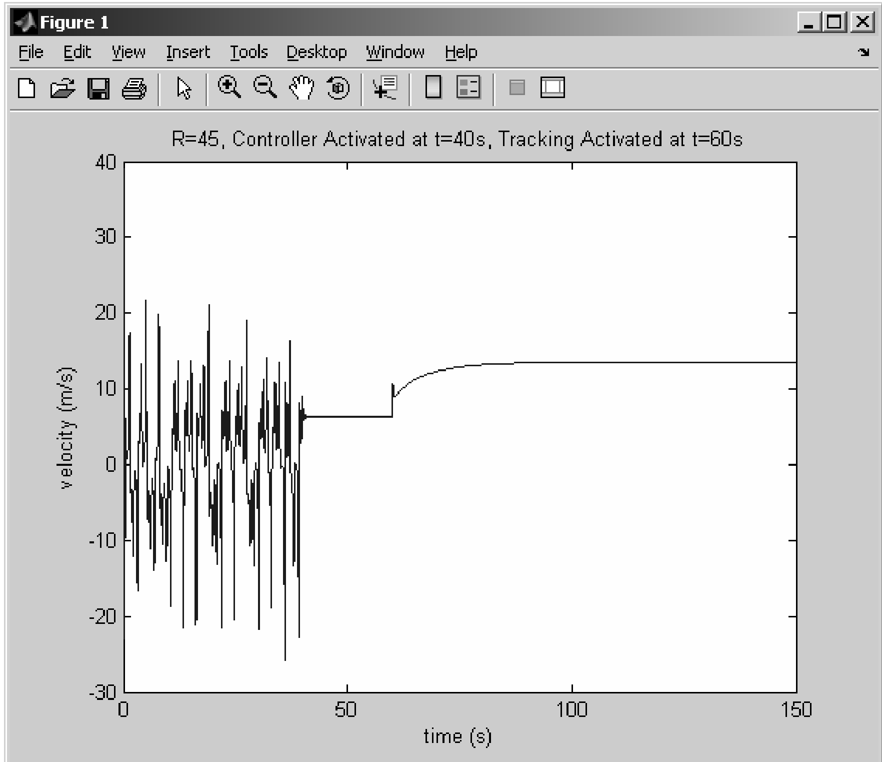

Figure 6.1 Plot of x vs. time with Quotient Controller and Tracking at R=45

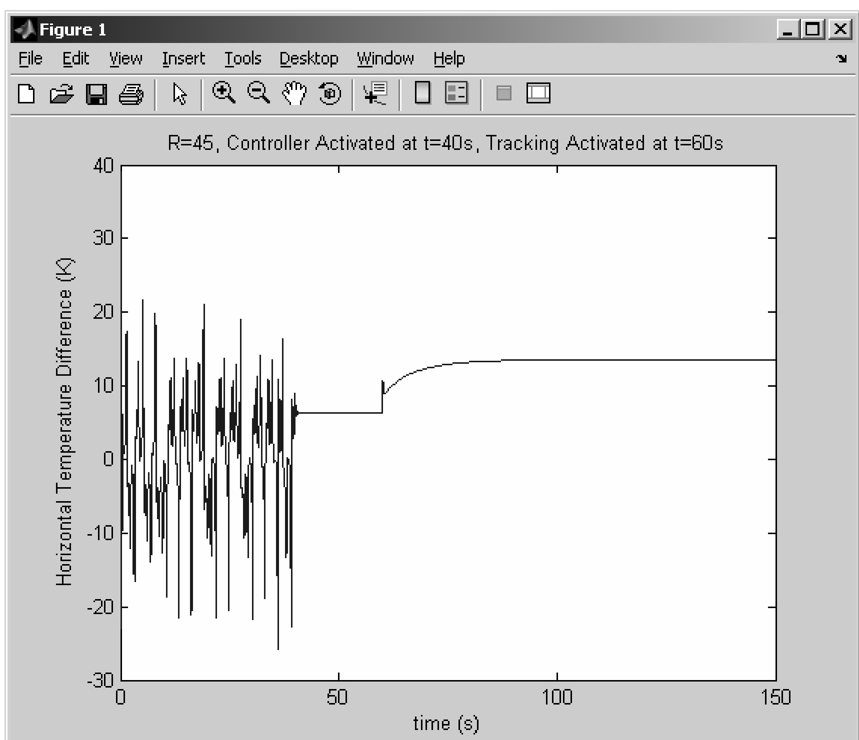

Figure 6.2 Plot of y vs. time with Quotient Controller and Tracking at R=45

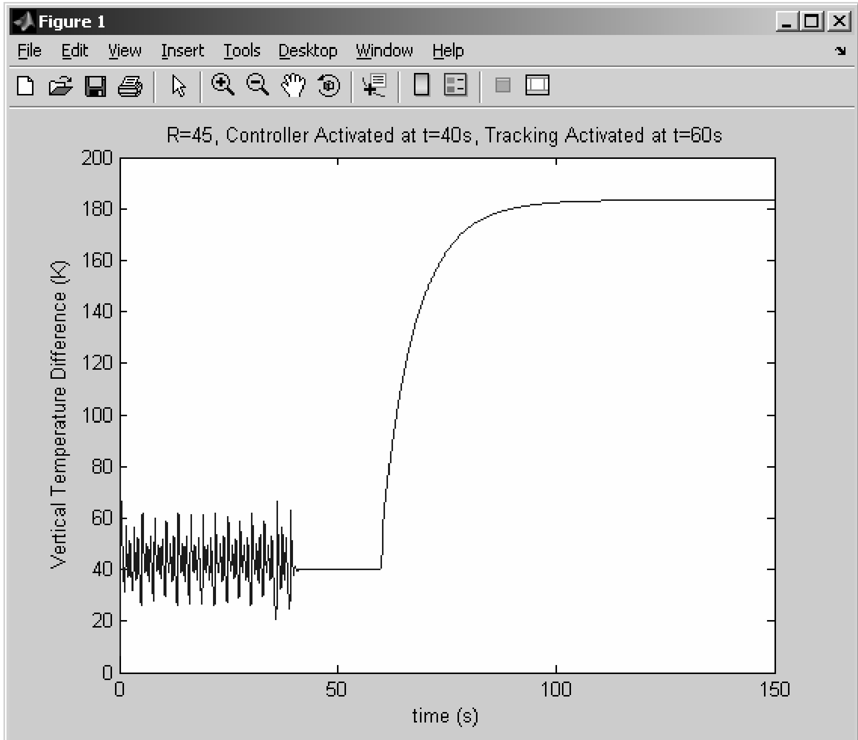

Figure 6.3 Plot of z vs. time with Quotient Controller and Tracking at R=45

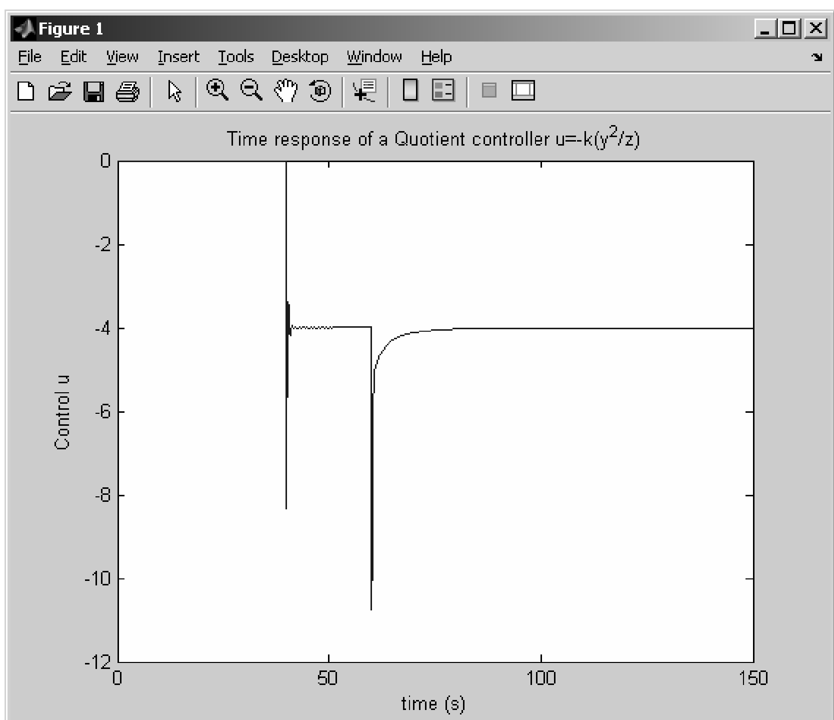

Figure 6.4 Plot of Quotient Controller vs. time with Controller and Tracking at R=45

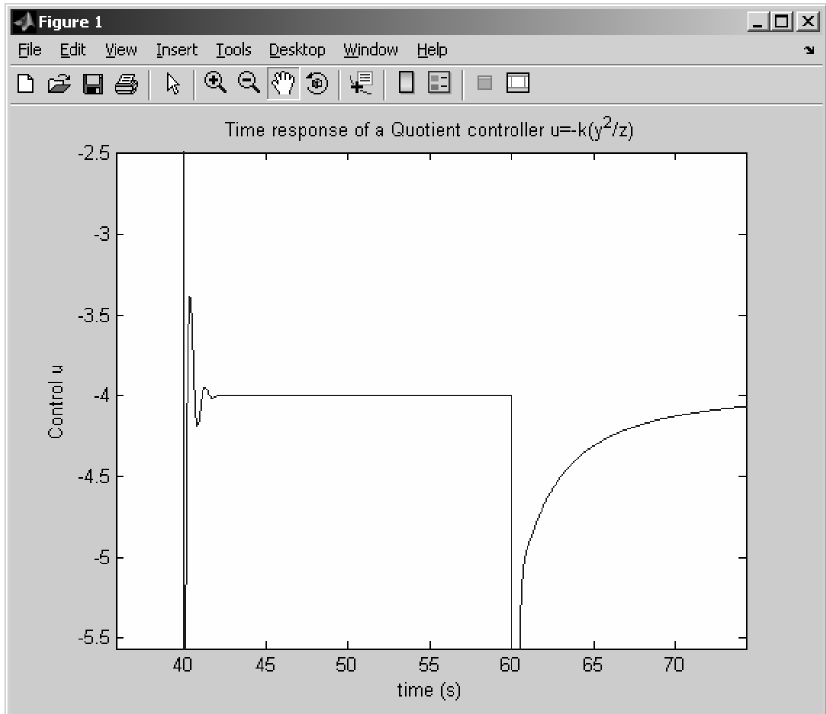

Figure 6.5 Detail of Quotient Controller vs. time with Controller and Tracking at R=45

To gain a better understanding of the behavior of the system, it is useful to examine how the quotient controller changes over time. In Figure 6.4, it is easy to see that initially the value of the quotient controller is zero because the controller is not activated until t = 40s. At t = 40s, we can see that the quotient controller immediately turns on and starts to oscillate until stability is achieved. Once stability is achieved, the value of the quotient controller remains constant until the tracking integrator is turned on (approximately at t = 60s) and the system is destabilized. After the system is destabilized by adding the tracking integrator, it quickly goes to the value that is being tracked to. It is difficult to see the more detailed behavior of the quotient controller in Figure 6.4, so a more detailed plot showing the small changes in the quotient controller from t = 40s to t = 60s is shown in Figure 6.5.

CHAPTER 7

CONCLUSION

Summary

The single-loop thermosyphon can be used almost anywhere there is a system that needs to have some heat removed. A specific example of this is in the core of nuclear reactors. The single-loop thermosyphon is a very efficient method of removing heat because it has no internal moving parts and solely depends on the convective flow of the fluid within the loop to remove heat. We have shown both analytically and through numerical simulation in MATLAB that for values of R within a specific range the single-loop thermosyphon system is stable and there is a steady convective fluid flow within the thermosyphon loop. Thus we have an effective means of cooling.

The problem arose when the value of R for the system was beyond the range that would normally lead to steady, convective flow. In this case, the thermosyphon system without a controller will behave in a chaotic manner. By introducing the quotient controller, we are able to maintain a stable, convective flow well within the normally chaotic range of R values. The quotient controller has two very important benefits over other controllers. The first beneficial feature is that the quotient controller is inexpensive to implement. For values of R that are well within the chaotic range, the value for the gain of the quotient controller remains small. For example, when $R=30$ the value for the gain of the quotient controller is less than 1 ($k<1$). Another benefit of the structure of the quotient controller is that the values of the variables that are used to describe the quotient controller are easily obtained from the real world system. The only variables that are used to construct the quotient controller, other than the gain, k, are the horizontal and vertical temperature difference of the loop. Since the only variables used to define the quotient controller are temperatures that can be easily obtained from measuring the system, the quotient controller can be easily implemented.

After showing that stability can be returned to the chaotic system with a relatively large value of R, we examined the interesting and useful feature of adding a tracking integrator to the system. The tracking integrator is used to allow the system the "track" to a specific temperature of velocity of the fluid within the single-loop thermosyphon. We have shown through simulation that with the tracking integrator added to the system we can maintain the stability of the system and track to a specific value for values of R that would normally lead to a chaotic fluid flow. Having a stable system means that there is a stable, convective flow within the loop, and we have an effective means of cooling.

Future Work

Possible future work could include applying the quotient controller and tracking integrator examined in this thesis to a multiple-loop system. The research performed here only examined the stability of a single-loop thermosyphon. In order to consider a multiple-loop thermosyphon system, we would need to include coupling terms to take into account the behavior of the fluid flow and heat flow at the connection between the individual loops.

Other future work could include examining the non-existence of periodic orbits. Through the simulation done, we have not observed any periodic orbits. However, we have not shown analytically that we will never observe periodic orbits for the single-loop thermosyphon with a quotient controller and tracking. More work could be done in trying to prove the non-existence of periodic orbits.

REFERENCES

[1] Stephen H. Davis and Michael N. Roppo, *Coupled Lorenz Oscillators*, Physica 24D, 1987

[2] P. Ehrhad, Ch. Karcher, and U. Müller, *Dynamical Behavior of Natural Convection in a Double-Loop System*, Experimental Heat Transfer, vol. 2, 1989

[3] Laurene V. Fausett, *Numerical Methods: Algorithms and Applications,* 2003

[4] G.H. Hostetter, C.J. Savant, and R.T. Stefani, *Design of Feedback Control Systems*, CBS College Publishing, 1982

[5] Kincaid and Cheney, *Numerical Analysis: Mathematics of Scientific Computing, 3rd Edition*, 2001

[6] Teh-Lu Liao and Sheng-Hung Lin, *Adaptive Control and Synchronization of Lorenz Systems*, Journal of the Franklin Institute 336 (1999)

[7] Süli and Mayers, *An Introduction to Numerical Analysis*, 2003

[8] Young and Okiishi, *Fundamentals of Fluid Mechanics, 5th Edition*, 2005

APPENDIX A

NOMENCLATURE

b	Biot number
l	radius of single-loop thermosyphon
$u(t)$	cross-sectional average velocity of fluid within the loop
ϕ	polar angle
ρ_0	density of fluid at temperature T_0
p	Prandtl number
P	pressure within the loop
g	acceleration due to gravity
$\rho(T)$	density of the fluid
f_w	frictional force acting on wall of loop
c_p	specific heat at constant pressure
$T(\phi,t)$	cross-sectional average temperature
h_w	heat transfer coefficient at wall of loop
$T_w(\phi)$	temperature distribution imposed on loop wall
f_{w0}	coefficient of friction at loop wall
ΔT	temperature difference (T_H-T_C)
T_H	constant wall temperature in lower half of loop
T_C	constant wall temperature in upper half of loop
C_n	Fourier coefficient for vertical temperature difference
S_n	Fourier coefficient for horizontal temperature difference
α_0	coefficient of thermal expansion
R	Rayleigh number, referring to the heat index imposed at the bottom of the loop
T_0	reference temperature
x	represents velocity of fluid within loop
y	represents horizontal temperature difference
z	represents vertical temperature difference

APPENDIX B

MATLAB CODE FOR SIMULATIONS

Code for simulation of single-loop thermosyphon

```
%Thermosyphon Simulation Main Program
%Name: Jonathan S. Tanner
%
%Date: July 18, 2006
%-------------------------------------------------------------
%This program calls several sub-routines that perform a simulation
%   of the behavior of a single-loop thermosyphon.
%Variables:
%   R -- Rayleigh number
%   p -- Prandtl number
%   x -- velocity of fluid inside loop
%   y -- horizontal temperature difference
%   z -- vertical temperature difference
%   s -- state vector s=[x,y,z]
%   t -- time vector
%Functions called:
%   rk() -- performs numerical integration using 4th-order R-K method

clear all
close all
format short e

%define the values for Prandtl and Rayleigh numbers
p = 10;
R = 10;

%define Steady-State solutions for Lorenz eqns for Thermosyphon
%ss=[0,0,0]; %trivial solution
ss=[sqrt(R-1),sqrt(R-1),R-1];
%ss=[-sqrt(R-1),-sqrt(R-1),R-1];

%define initial conditions
ic = [ss(1)+0.01, ss(2)+0.01, ss(3)+0.01];
%ic = [0.1, 0.1, 1];

%define current state vector s = [x,y,z]
s = ic;

%input values for numerical integration interations
%maxi = input('Enter maximum number of iterations:');
maxi = 5000;
%h = input('Enter value of step size for R-K method:');
h = 0.01;

%call function to perform 4th-order Runge-Kutta method
s = rk(maxi, h, p, R, s);

%define time vector
t = [0:h:(maxi-1)*h]';

%plot velocity vs. time
subplot(2,2,1)
plot(t,s(:,1))
title('Velcoity vs. Time')
```

```
xlabel('time')
ylabel('velocity')

%phase plot (velocity vs. horizontal temp difference)
subplot(2,2,2)
plot(s(:,1),s(:,2))
title('Phase plot')
xlabel('velocity')
ylabel('Horizontal temp. difference')

%plot horizontal temp. difference
subplot(2,2,3)
plot(t,s(:,2))
title('Horizontal temp. difference vs. Time')
xlabel('time')
ylabel('Horizontal temp. difference')

%plot vertical temp. difference
subplot(2,2,4)
plot(t,s(:,3))
title('Vertical temp. difference vs. Time')
xlabel('time')
ylabel('Vertical temp. difference')
```

Code for simulation of single-loop thermosyphon that creates bifurcation plots

```
%Thermosyphon Bifurcation Plot Main Program
%Name: Jonathan S. Tanner
%
%Date: July 20, 2006
%------------------------------------------------------------
%This program calls several sub-routines that perform a simulation
%   of the behavior of a single-loop thermosyphon and create a
%   bifurcation plot using the simulation data.
%Variables:
%   R -- Rayleigh number
%   p -- Prandtl number
%   x -- velocity of fluid inside loop
%   y -- horizontal temperature difference
%   z -- vertical temperature difference
%   s -- state vector s=[x,y,z]
%   t -- time vector
%   maxi -- max number of iteration
%   h -- step size for iterations
%Functions called:
%   rk() -- performs numerical integration using 4th-order R-K method

clear all
close all
format short e

%define the values for Prandtl and Rayleigh numbers
p = 10;
R = [2:0.5:17];

N = length(R);

%define Steady-State solutions for Lorenz eqns for Thermosyphon
%ss=[0,0,0]; %trivial solution
ss=[sqrt(R-1),sqrt(R-1),R-1];
```

```
%ss=[-sqrt(R-1),-sqrt(R-1),R-1];

%define initial conditions
ic = [ss(1)+0.01, ss(2)+0.01, ss(3)+0.01];

for k=1:N
   R0 = R(k);

   %define current state vector s = [x,y,x]
   s = ic;

   %perform interation (numerical integration)using Runge-Kutta
   %   4th-order method & return state vector
   maxi = 2000; h=0.01;
   s = rk(maxi, h, p, R0, s);

   %remove transient part (take only last value of state vector)
   s = s(maxi, :);

   %perform interations using new state vector as initial condition
   maxi = 3000; h=0.01;
   s = rk(maxi, h, p, R0, s);


   %find and plot peak values for x
   subplot(3,1,1)
   %hold on so that subsequent graphing commands add to the existing graph
   hold on
   for i=1:maxi-2
      if(s(i,1)<s(i+1,1) & s(i+1,1)>s(i+2,1))
         %peak
         plot(R0,s(i+1,1))
      %elseif(s(i,1)>s(i+1,1) & s(i+1,1)<s(i+2,1))
         %trough
         %plot(R0,s(i+1,1))
      end
      pause(0.01)
   end

   %find and plot peak values for y
   subplot(3,1,2)
   hold on
   for i=1:maxi-2
      if(s(i,2)<s(i+1,2) & s(i+1,2)>s(i+2,2))
         %peak
         plot(R0,s(i+1,2))
      %elseif(s(i,2)>s(i+1,2) & s(i+1,2)<s(i+2,2))
         %trough
         %plot(R0,s(i+1,2))
      end
      pause(0.01)
   end

   %find and plot peak values for z
   subplot(3,1,3)
   hold on
   for i=1:maxi-2
      if(s(i,3)<s(i+1,3) & s(i+1,3)>s(i+2,3))
         %peak
         plot(R0,s(i+1,3))
      %elseif(s(i,3)>s(i+1,3) & s(i+1,3)<s(i+2,3))
         %trough
```

```
         %plot(R0,s(i+1,3))
      end
      pause(0.01)
   end

end

subplot(3,1,1)
title('Bifurcation Plot (x vs. R)')
xlabel('R')
ylabel('x')

subplot(3,1,2)
title('Bifurcation Plot (y vs. R)')
xlabel('R')
ylabel('y')

subplot(3,1,3)
title('Bifurcation Plot (z vs. R)')
xlabel('R')
ylabel('z')

%hold off so that PLOT commands erase the previous plots
hold off
```

M-files for simulation of single-loop thermosyphon without quotient controller

```
%M-file: rk.m
%
%This function performs performs numerical integration
%   using 4th-order Runge-Kutta method.
%
%  rk(maxi, h, p, R, s)
%
%Variables:
%   maxi-- number of interations
%   h   -- step size
%   p   -- Prandtl number
%   R   -- Rayleigh number
%   s   -- state vector s = [x,y,z]
%Functions called:
%   lorenz()  -- calls lorenz equation that govern thermosyphon

function new_s = rk(maxi, h, p, R, s)

for k=1:maxi

   F1 = lorenz(s, p, R);
   F2 = lorenz(s+h/2*F1,p,R);
   F3 = lorenz(s+h/2*F2,p,R);
   F4 = lorenz(s+h*F3,p,R);

   s = s+h/6*(F1+2*F2+2*F3+F4);

   new_s(k,1:3) = s;
end

------------------------------------------------------------------------------
```

```
%M-file: lorenz.m
%
%This file defines function that defines Lorenz equations
%   that govern the thermosyphon.
%
%   lorenz(s,p,R)
%
%Variables
%   s -- state vector s = [x,y,z]
%   p -- Prandtl number
%   R -- Rayleigh number

function sdot = lorenz(s,p,R)

%define x, y, z
x=s(1); y=s(2); z=s(3);

%equation for velocity
xp = p*(y-x);

%equation for hoizontal temp. difference
yp = R*x-y-x*z;

%equation for vertical temp. difference
zp = x*y-z;

sdot = [xp,yp,zp];
```

Code for simulation of single-loop thermosyphon with quotient controller

```
%Thermosyphon Controller Simulation Main Program
%Name: Jonathan S. Tanner
%
%Date: September 7, 2006
%--------------------------------------------------------------
%This program calls several sub-routines that perform a simulation
%   of the behavior of a single-loop thermosyphon with quotient controller.
%Variables:
%   R -- Rayleigh number
%   p -- Prandtl number
%   x -- velocity of fluid inside loop
%   y -- horizontal temperature difference
%   z -- vertical temperature difference
%   s -- state vector s=[x,y,z]
%   t -- time vector
%   k -- value for gain of quotient controller
%Functions called:
%   rk_cont() -- performs numerical integration using 4th-order R-K method

clear all
close all
format short e

%define the values for Prandtl and Rayleigh numbers
```

```
p = 10;
R = 40;
%define value for gain of quotient controller
%  need to add a small value to k to account for approximation error
%  and machine accuracy
k =((-280-2*R+sqrt(4*R^2+2400*R+56000))/80)+0.1;

%define Steady-State solutions for Lorenz eqns for Thermosyphon
%ss=[0,0,0]; %trivial solution
ss=[sqrt(R-1),sqrt(R-1),R-1];
%ss=[-sqrt(R-1),-sqrt(R-1),R-1];

%define initial conditions
ic = [ss(1)+0.01, ss(2)+0.01, ss(3)+0.01];
%ic = [0.1, 0.1, 1];

%define current state vector s = [x,y,z]
s = ic;

%input values for numerical integration interations
%maxi = input('Enter maximum number of iterations:');
maxi = 5000;
%h = input('Enter value of step size for R-K method:');
h = 0.01;

%call function to perform 4th-order Runge-Kutta method
s = rk_cont(maxi, h, p, R, s, k);

%define time vector
t = [0:h:(maxi-1)*h]';

%plot velocity vs. time
subplot(2,2,1)
plot(t,s(:,1))
title('Velcoity vs. Time')
xlabel('time')
ylabel('velocity')

%phase plot (velocity vs. horizontal temp difference)
subplot(2,2,2)
plot(s(:,1),s(:,2))
title('Phase plot')
xlabel('velocity')
ylabel('Horizontal temp. difference')

%plot horizontal temp. difference
subplot(2,2,3)
plot(t,s(:,2))
title('Horizontal temp. difference vs. Time')
xlabel('time')
ylabel('Horizontal temp. difference')

%plot vertical temp. difference
subplot(2,2,4)
plot(t,s(:,3))
title('Vertical temp. difference vs. Time')
xlabel('time')
ylabel('Vertical temp. difference')
```

M-files for simulation of single-loop thermosyphon with quotient controller

```
%M-file: rk_cont.m
%
%This function performs numerical integration
%   using 4th-order Runge-Kutta method.
%
%  rk_cont(maxi, h, p, R, s, k)
%
%Variables:
%   maxi-- number of interations
%   h   -- step size
%   p   -- Prandtl number
%   R   -- Rayleigh number
%   s   -- state vector s = [x,y,z]
%   k   -- gain factor for quotient controller
%Functions called:
%   lorenz_cont()  -- calls lorenz equations that govern thermosyphon

function new_s = rk_cont(maxi, h, p, R, s, k)

for n=1:maxi

%turn on the controller after a set number of iterations
if n>(maxi/5)
   gain = k;
else
   gain = 0;
end

   F1 = lorenz_cont(s, p, R, gain);
   F2 = lorenz_cont(s+h/2*F1,p,R, gain);
   F3 = lorenz_cont(s+h/2*F2,p,R,gain);
   F4 = lorenz_cont(s+h*F3,p,R,gain);

   s = s+h/6*(F1+2*F2+2*F3+F4);

   new_s(n,1:3) = s;
end
```

```
%M-file: lorenz_cont.m
%
%This file defines function that defines Lorenz equations
%   that govern the thermosyphon.
%
%   lorenz_cont(s,p,R, k)
%
%Variables
%   s -- state vector s = [x,y,z]
%   p -- Prandtl number
%   R -- Rayleigh number
%   k -- gain factor for quotient controller
%   u -- quotient contoller

function sdot = lorenz_cont(s,p,R,k)

%define x, y,z
```

```
x=s(1); y=s(2); z=s(3);

%equation for velocity
xp = p*(y-x);

%equation for quotient controller
u = -k*y^2/z;

%equation for hoizontal temp. difference
yp = (R+u)*x-y-x*z;

%equation for vertical temp. difference
zp = x*y-z;

sdot = [xp,yp,zp];
```

Code for simulation of single-loop thermosyphon with tracking

```
%       Date 02/6/2007
%       Global stability of controlling Lorenz equations
%       Special controll structure: u=-ky^2/z
%       tracking an input signal, e.g. a step input.  A new DE is incorporated into the system:
%       wdot=y-yr
%       new controller: u=-ky^2/z-P(y-yr)-Lw(y-yr), here Lw acts like an integrator (PI-controller)
        clear all
        close all
        format short e
        P=10; % Prandtel number
        R=45;  % Rayleigh number
        b=1;  % Biot number nominal value is 8/3
        ss=[0,0,0];  % steady states
%       ss=[sqrt(b*(R-1)),sqrt(b*(R-1)),R-1];  % Steady states
%       ss=[-sqrt(b*(R-1)),-sqrt(b*(R-1)),R-1];  % Steady states


        Gain1=4;  % the gain for the quotient controller
        Gain2=3;  % the gain for the integrator

        T=0.01;  % simulation step size
        startc=40;  % when to activate the controller (in seconds)

        Sindx1=startc/T;      % start to stabilize the system
        Sindx2=(startc+20)/T; % start to track the input

        loops=15000;   %  simulation steps
%       Initial=[ss(1)-6,ss(2)+8,ss(3)+5]; % initial conditions

%       Initial=[-10 11.4444476274911766 12];    % WOW WOW WOW, what is the limit!!!!!
    Initial=[5 13 7 0];  % the fourth state is the integrator

%       Initial=[.1,.1,.5];
        u=0;  % control input
        state=Initial;
        stdata1=state(2); % record the y-output (can also record other states)
        stdata2=state(3); % record the z-output
        Cinput=u;       % record the control signal

        beta=0;
        target=13.54;  % input signal to track
```

```
        para=[P,R,b,beta,Gain1,Gain2,target];  % fourth element is beta=1 means activate tracking
controller
        rec_state=Initial;
        for k=1:loops
          if k>=Sindx1 & k<Sindx2
                u=-Gain1*state(2)^2/state(3);   % control input u=-k(y^2/z)
          end

          if k>=Sindx2
                beta=1;
                para(4)=beta;  % fourth element is beta=1 means activate tracking controller
    u=-Gain1*state(2)^2/state(3);   % control input u=-k(y^2/z)
    end



          state=csimlor2(state,para,u,T);
          stdata1=[stdata1;state(2)]; % record the y-output (can also record other states)
          stdata2=[stdata2;state(3)]; % record the z-output

          Cinput=[Cinput;u];
          rec_state=[rec_state;state];
        end
        time=[0:T:loops*T];
        figure(1)
        subplot(3,1,1)
        plot(time,stdata1)
        title('Time responses of y&z-output of Lorenz equations')
        ylabel('Y')
        subplot(3,1,2)
        plot(time,stdata2)
        ylabel('Z')
        subplot(3,1,3)
        plot(time,Cinput)
        title('Time response of a Quotient controller u=-k(y^2/z)')
        ylabel('Control u')

 x=rec_state(:,1);
 y=rec_state(:,2);
 z=rec_state(:,3);

 figure(2)
 plot3(x,y,z,'r-')
```

M-files for simulation of single-loop thermosyphon with quotient controller

```
%       Date 02/06/2007
%       This function is set up for the system of ODEs
%       for the Lorenz system.   All system
%       parameters are defined within these function.
%
%       Note:  the input data is  a matrix, the  ith row
%       represent the  ith loop,  and the Three elements
%       in  the  ith row represent Velocity, Cosine, and
%       Sine  coefficients  of the temperature variable.
%       The  output  of this function  is  also a matrix
%       which has the same structure as the input matrix
%       with entries  as  the  derivatives  of the state
%       variables.
```

```
          function Sdot=clorsys2(State,para,u)
%         System parameters
          P=para(1);  % Prandtle number
          R=para(2);  % Rayleigh number
          b=para(3);  % Biot number
      beta=para(4);
          gain1=para(5);
          gain2=para(6);
          yr=para(7);

          X=State(1);   Y=State(2);   Z=State(3);  W=State(4);
          tracku_I=-gain1*W;           % integrator
          tracku_p=-gain2*beta*(Y-yr);  % proportional control

%         **************************************************
          Xdot=-P*X+P*Y;  % the ODEs or state equations
          Ydot=(R+u+tracku_p+tracku_I)*X-X*Z-Y;
          Zdot=-b*Z+X*Y;
          Wdot=beta*(Y-yr);
%         **************************************************
          Sdot=[Xdot,Ydot,Zdot,Wdot];

-----------------------------------------------------------------------------------------------------------------
%         Date 08/13/99
%         This function is used to  perform   numerical  integration
%         of the ODE.  Normally, we use one   function   sample  and
%         one or more  than  one  derivative  samples.     The input
%         matrix should follow the format:   the size of  the matrix
%         is 3 by M, the three rows, the data represent Velocity,Cosine,
%         and Sine coefficients of the temperature  variable.   In  each
%         row, the first column is the function sample, and the rest
%         of the columns are the derivative samples.
%         We can use linear prediction  schemes  with  correctors or
%         modifiers.
%
%         Note: the derivative samples should be   arranged from the
%              oldest to the latest. Then they will be conversed in
%              this subroutine.
%
%         Note: the simulation stepsize is given  as an input parameter
%
%         Note: the output is a single vector with the predicted values
%              of the states



          function Newst=csimlor2(State,para,u,T)
%         4th order Runge-Kutta method
          K1=clorsys2(State,para,u);
          K2=clorsys2(State+T/2*K1,para,u);
          K3=clorsys2(State+T/2*K2,para,u);
          K4=clorsys2(State+T*K3,para,u);
       Newst=State+T/6*(K1+2*K2+2*K3+K4);
```

APPENDIX C

MAPLE OUTPUT FOR STABILITY ANALYSIS

clear all variables

> $x := 'x' : y := 'y' : z := 'z' : w := 'w' : d := 'd' : k := 'k' :$

> $with(linalg) :$

> $$J := \left[[-10, 10\ , 0, 0], [(R-k-d\cdot w-l\cdot(y-yr)-z), (-2\cdot k —l\cdot x-1), \left(\frac{k}{x}-x\right), -d\cdot\ x], [x, x, -1, 0], [0, 1, 0, 0] \right];$$

$$J := \begin{bmatrix} -10 & 10 & 0 & 0 \\ R-k-d\,w-l\,(y-1)-z & -2\,k+l\,x-1 & \frac{k}{x}-x & -d\,x \\ x & x & -1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

> $JL := \left[[-10-\lambda, 10\ \ , 0, 0], [(R-k-d\cdot w-l\cdot(y-yr)-z), (-2\cdot k-l\cdot x-1)-\lambda, \left(\frac{k}{x}-x\right), -d\cdot x], [x, x, -1-\lambda, 0], [0, 1, 0, 0-\lambda] \right];$

$$JL := \begin{bmatrix} -10-\lambda & 10 & 0 & 0 \\ R-k-d\,w-l\,(y-1)-z & -2\,k-l\,x-1-\lambda & \frac{k}{x}-x & -d\,x \\ x & x & -1-\lambda & 0 \\ 0 & 1 & 0 & -\lambda \end{bmatrix}$$

> $x := y:$

> $collect(det(JL), \lambda);$

$$\lambda^4 + (2\,k + 12 + l\,y)\,\lambda^3 + \left(21\,l\,y - 10\,R - 10\,l + 10\,d\,w + 31\,k + d\,y + 10\,z + 21 + y^2\right)\lambda^2 + \left(10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k\right)\lambda + 10\,d\,y$$

> $a := 1;\ b := (12 + 2\,k + l\cdot y);\ c := \left(10\,z + 10\,d\,w + y^2 + 21\,l\,y + 21 - 10\,R - 10\,l\,yr + 31\,k + d\,y\right);$

$$a := 1$$

$$b := 2\,k + 12 + l\,y$$

$$c := 21\,l\,y - 10\,R - 10\,l + 10\,d\,w + 31\,k + d\,y + 10\,z + 21 + y^2$$

> $e := \left(-10\,R - 10\,l\,yr + 11\,d\,y + 10\,z + 20\,l\,y + 10\,d\,w + 20\,y^2 + 10\,k + 10\right);\ f := 10\,d\,y;$

$$e := 10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k$$

$$f := 10\,d\,y$$

> $linalg[matrix](2, 2, [a, c, b, e]);$

$$[1, 21\,l\,y - 10\,R - 10\,l + 10\,d\,w + 31\,k + d\,y + 10\,z + 21 + y^2] \\ 2\,k + 12 + l\,y, \\ 10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k]$$

> $a1 := \frac{-det(\%)}{b};$

$$a1 := -\frac{1}{2\,k + 12 + l\,y}\big(-242 + 110\,R + 110\,l - 110\,z - 404\,k + 8\,y^2 + 10\,R\,l\,y - 20\,d\,w\,k - d\,y^2\,l - 10\,z\,l\,y - 73\,l\,y\,k - 2\,d\,y - 110\,d\,w + 20\,l\,k + 10\,l^2\,y - 20\,z\,k - 2\,y^2\,k - y^3\,l - 21\,l^2\,y^2 + 20\,R\,k - 62\,k^2 - 10\,d\,w\,l\,y - 253\,l\,y - d\,y\big)$$

> $solve(a1 = 0, d);$

$$\frac{1}{y^2\,l + 110\,w + 2\,y\,k + 20\,w\,k + 10\,w\,l\,y + y}\big(-242 + 110\,R + 110\,l - 110\,z - 404\,k + 8\,y^2 + 10\,R\,l\,y - 62\,k^2 - 2\,y^2\,k - 10\,z\,l\,y - 73\,l\,y\,k - 20\,z\,k + 20\,l\,k + 10\,l^2\,y + 20\,R\,k - y^3\,l - 21\,l^2\,y^2 - 253\,l\,y\big)$$

> $linalg[matrix](2, 2, [c, f, e, 0]);$

$$[21\,l\,y - 10\,R - 10\,l + 10\,d\,w + 31\,k + d\,y + 10\,z + 21 + y^2, \\ 10\,d\,y], [\\ 10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k, 0]$$

> $a6 := \frac{-det(\%)}{b};$

$$a6 := \frac{1}{2\,k + 12 + l\,y}\big(10\,d\,y\,(10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k)\big)$$

> $a2 := a6;$

$$a2 := \frac{1}{2\,k + 12 + l\,y}\,\big(10\,d\,y\,\big(10 + 10\,z + 11\,d\,y + 10\,d\,w + 20\,l\,y - 10\,l + 20\,y^2 - 10\,R + 10\,k\big)\big)$$

> $solve(a6 = 0, d);$

$$0, \frac{10\,\big(R + l\,yr - z - 2\,l\,y - 2\,y^2 - k - 1\big)}{11\,y + 10\,w}$$

> $linalg[matrix](2, 2, [b, e, a1, a2]);$

$$[2\,k + 12 + l\,y, -10\,R - 10\,l\,yr + 11\,d\,y + 10\,z + 20\,l\,y + 10\,d\,w + 20\,y^2 + 10\,k + 10], [-\frac{1}{2\,k + 12 + l\,y}\,\big(-242 + 110\,R - 110\,z - 404\,k + 110\,l\,yr + 8\,y^2 + 10\,l^2\,yr\,y + 10\,R\,l\,y - 20\,d\,w\,k - d\,y^2\,l + 20\,l\,yr\,k - 10\,z\,l\,y - 73\,l\,y\,k - 2\,d\,y\,k - 110\,d\,w - 20\,z\,k - 2\,y^2\,k - y^3\,l - 21\,l^2\,y^2 + 20\,R\,k - 62\,k^2 - 10\,d\,w\,l\,y - 253\,l\,y - d\,y\big), \frac{1}{2\,k + 12 + l\,y}\,\big(10\,d\,y\,\big(-10\,R - 10\,l\,yr + 11\,d\,y + 10\,z + 20\,l\,y + 10\,d\,w + 20\,y^2 + 10\,k + 10\big)\big)]$$

> $a4 := \frac{-det(\%)}{a1};$

$$a4 := -\Big(\big(10\,R + 10\,l\,yr - 11\,d\,y - 10\,z - 20\,l\,y - 10\,d\,w - 20\,y^2 - 10\,k - 10\big)\,\big(-242 + 110\,R - 110\,z - 404\,k + 110\,l\,yr + 8\,y^2 + 10\,l^2\,yr\,y + 10\,R\,l\,y - 20\,d\,w\,k + 9\,d\,y^2\,l + 20\,l\,yr\,k - 10\,z\,l\,y - 73\,l\,y\,k + 18\,d\,y\,k - 110\,d\,w - 20\,z\,k - 2\,y^2\,k - y^3\,l - 21\,l^2\,y^2 + 20\,R\,k - 62\,k^2 - 10\,d\,w\,l\,y - 253\,l\,y + 119\,d\,y\big)\Big) \Big/ \big(-242 + 110\,R - 110\,z - 404\,k + 110\,l\,yr + 8\,y^2 + 10\,l^2\,yr\,y + 10\,R\,l\,y - 20\,d\,w\,k - d\,y^2\,l + 20\,l\,yr\,k - 10\,z\,l\,y - 73\,l\,y\,k - 2\,d\,y\,k - 110\,d\,w - 20\,z\,k - 2\,y^2\,k - y^3\,l - 21\,l^2\,y^2 + 20\,R\,k - 62\,k^2 - 10\,d\,w\,l\,y - 253\,l\,y - d\,y\big)$$

> $solve(a4 = 0, d);$

$$\frac{10\left(R + l\,yr - z - 2\,l\,y - 2\,y^2 - k - 1\right)}{11\,y + 10\,w},$$
$$-\frac{1}{-20\,w\,k + 9\,y^2\,l + 18\,y\,k - 110\,w - 10\,w\,l\,y + 119\,y} (-242$$
$$+ 110\,R - 110\,z - 404\,k + 110\,l\,yr + 8\,y^2 + 10\,l^2\,yr\,y$$
$$+ 10\,R\,l\,y + 20\,l\,yr\,k - 10\,z\,l\,y - 73\,l\,y\,k - 20\,z\,k - 2\,y^2\,k - y^3$$
$$-21\,l^2\,y^2 + 20\,R\,k - 62\,k^2 - 253\,l\,y)$$

define steady-state values of x, y, z, w

> $x := yr;\ y := yr;\ z := yr^2;\ w := \frac{(R - k - 1 - yr^2)}{d};$

$$x := yr$$
$$y := yr$$
$$z := yr^2$$
$$w := \frac{R - k - 1 - yr^2}{d}$$

> $a1;$

$$-\frac{1}{2\,k + 12 + l\,yr} (-132 - 294\,k - 143\,l\,yr$$
$$+ 8\,yr^2 - 22\,yr^2\,k - d\,yr - 20\left(R - k - 1 - yr^2\right)k - 53\,l\,yr\,k$$
$$-10\left(R - k - 1 - yr^2\right)l\,yr$$
$$+ 10\,R\,l\,yr - 11\,l^2\,yr^2 - d\,yr^2\,l - 11\,yr^3\,l - 2\,d\,yr\,k$$
$$+ 20\,R\,k - 62\,k^2)$$

> $solve(a1 = 0, d);$

$$-\frac{1}{yr\,(1 + 2\,k + l\,yr)} (132 + 274\,k + 133\,l\,yr - 8\,yr^2 + 2\,yr^2\,k$$
$$+ 42\,k^2 + 43\,l\,yr\,k + yr^3\,l + 11\,l^2\,yr^2)$$

> $a6;$

$$\frac{10\,d\,yr\left(10\,l\,yr + 11\,d\,yr + 20\,yr^2\right)}{2\,k + 12 + l\,yr}$$

> $solve(a6 = 0, d);$

$$0,\ -\frac{10}{11}\,l - \frac{20}{11}\,yr$$

> $a4;$

$$-\left(\left(-10\,l\,yr-11\,d\,yr-20\,yr^2\right)\left(-132-294\,k-143\,l\,yr+8\,yr^2-22\,yr^2\,k+119\,d\,yr-20\left(R-k-1-yr^2\right)k-53\,l\,yr\,k-10\left(R-k-1-yr^2\right)l\,yr+10\,R\,l\,yr-11\,l^2\,yr^2+9\,d\,yr^2\,l-11\,yr^3\,l+18\,d\,yr\,k+20\,R\,k-62\,k^2\right)\right)\Big/\left(-132-294\,k-143\,l\,yr+8\,yr^2-22\,yr^2\,k-d\,yr-20\left(R-k-1-yr^2\right)k-53\,l\,yr\,k-10\left(R-k-1-yr^2\right)l\,yr+10\,R\,l\,yr-11\,l^2\,yr^2-d\,yr^2\,l-11\,yr^3\,l-2\,d\,yr\,k+20\,R\,k-62\,k^2\right)$$

> $solve(a4=0,\,d);$

$$-\frac{10}{11}\,l-\frac{20}{11}\,yr,\ \frac{1}{yr\,(9\,l\,yr+119+18\,k)}\left(132+274\,k+133\,l\,yr-8\,yr^2+2\,yr^2\,k+42\,k^2+43\,l\,yr\,k+yr^3\,l+11\,l^2\,yr^2\right)$$

clear variables

> $k:='k';\ yr:='yr';\ d:='d';$

$$k:=k$$

$$yr:=yr$$

$$d:=d$$

define value for d to make system with integrator stable

> $d:=\frac{2\left(137\,k+66-4\,yr^2+21\,k^2+yr^2\,k\right)}{yr\,(119+18\,k)}-0.01;$

$$d:=\frac{274\,k+132-8\,yr^2+42\,k^2+2\,yr^2\,k}{yr\,(119+18\,k)}-0.01$$

define values for k and yr

> $k:=8.7494\cdot 10^{-1};\ yr:=1;$

$$k:=0.8749400000$$

$$yr:=1$$

> $a1;$

$$-\frac{1}{13.74988000+l}\left(-203.3127548\,l-405.7224979+10\,R\,l-11\,l^2-29.40934822\,(0.3400279369\,R-0.9775599169)\,l\right)$$

> *a4*;

$$-\left((-52.35028304-10\,l)\left(-173.9034066\,l+10\,R\,l-11\,l^2-29.40934822\,(0.3400279369\,R-0.9775599169)\,l-1.3474891\right)\right)\Big/\left(-203.3127548\,l-405.7224979+10\,R\,l-11\,l^2-29.40934822\,(0.3400279369\,R-0.9775599169)\,l\right)$$

> *a6*;

$$\frac{29.40934822\,(52.35028304+10\,l)}{13.74988000+l}$$

clear value of d

> *d* :='*d*';

$$d:=d$$

> $d := \frac{-2\left(137\,k+66-4\,yr^2+21\,k^2+yr^2\,k\right)}{yr\,(2\cdot k+1)}+0.02;$

$$d:=-144.5809572$$

> *a1*;

$$-\frac{1}{13.74988000+l}\left(1445.809572\,(-0.006916540182\,R+0.01988463803)\,l+10\,R\,l-55.7908628\,l-0.0549975-11\,l^2\right)$$

> *a4*;

$$-\left((1570.390529-10\,l)\left(1445.809572\,(-0.006916540182\,R+0.01988463803)\,l+10\,R\,l-1501.600435\,l-19879.76312-11\,l^2\right)\right)\Big/\left(1445.809572\,(-0.006916540182\,R+0.01988463803)\,l+10\,R\,l-55.7908628\,l-0.0549975-11\,l^2\right)$$

> *a6*;

$$-\frac{1445.809572\,(-1570.390529+10\,l)}{13.74988000+l}$$

> $d := \frac{-20}{11} \cdot yr + 0.01;$

$$d := -1.808181818$$

> $a1;$

$$-\frac{1}{13.74988000 + l} \big(10\,R\,l - 198.5636382\,l + 18.08181818\,(-0.5530417296\,R + 1.589961790)\,l - 11\,l^2 - 392.6629971\big)$$

> $a4;$

$$-\big((-0.110000000 - 10\,l)\,\big(10\,R\,l - 216.6454564\,l - 641.2858272 + 18.08181818\,(-0.5530417296\,R + 1.589961790)\,l - 11\,l^2\big)$$
$$\big(10\,R\,l - 198.5636382\,l + 18.08181818\,(-0.5530417296\,R + 1.589961790)\,l - 11\,l^2 - 392.6629971\big)$$

> $a6;$

$$-\frac{18.08181818\,(0.110000000 + 10\,l)}{13.74988000 + l}$$

> $d := 0 + 0.01;$

$$d := 0.01$$

> $a1;$

$$-\frac{1}{13.74988000 + l} \big(10\,R\,l - 200.3818200\,l - 0.10\,(100.\,R - 287.4940000)\,l - 397.6627789 - 11\,l^2\big)$$

> $a4;$

$$-\big((-20.11000000 - 10\,l)\,\big(10\,R\,l - 200.2818200\,l - 396.2877909 - 0.10\,(100.\,R - 287.4940000)\,l - 11\,l^2\big)\big) \big/ \big(10\,R\,l - 200.3818200\,l - 0.10\,(100.\,R - 287.4940000)\,l - 397.66277\ldots - 11\,l^2\big)$$

> $a6;$

$$\frac{0.10\,(20.11000000 + 10\,l)}{13.74988000 + l}$$

> $d := 'd':$

> $d := 0.1;$

$$d := 0.1$$

> *a1*;

$$-\frac{1}{13.74988000 + l}\left(10\,R\,l - 200.4718200\,l - 1.0\,(10.\,R - 28.74940000)\,l - 397.9102681 - 11\,l^2\right)$$

> *a4*;

$$-\left((-21.10000000 - 10\,l)\left(10\,R\,l - 199.4718200\,l - 384.1603881 - 1.0\,(10.\,R - 28.74940000)\,l - 11\,l^2\right)\right)\Big/\left(10\,R\,l - 200.4718200\,l - 1.0\,(10.\,R - 28.74940000)\,l - 397.9102681 - 11\,l^2\right)$$

> *a6*;

$$\frac{1.0\,(21.10000000 + 10\,l)}{13.74988000 + l}$$