

# **Leaky Waveguides: Experimental Characterization of Temperature, Wavelength, and Power Dependence**

Alexander Atanasiu, '05

Engineering Final Design Project Report-Draft  
*Submitted Tuesday, May 24, 2005*

## **i. Abstract**

A set of existing leaky waveguides were analyzed both analytically and experimentally. Analytic solutions were found and plotted for room temperature, for 1000 nm light, with the hope that these solutions will be extended to include temperature- and wavelength- dependent effects. A previous experiment was rebuilt to measure the temperature, wavelength, and power dependence of these waveguides. This setup was configured to cool and heat the waveguides automatically using a computer controlled resistance heater and thermoelectric cooler, and to run a laser control code (i.e. a laser power and wavelength scanner) once steady state had been achieved. This temperature control code was written in LabView version 6, and the laser controller VI was repaired and updated to work in this version to produce a seamless, fully automated experiment.

## ii. Acknowledgments

*This project could not have been completed without the patient guidance of Professor Lynne Molter. She introduced me to the project, and spent many long hours answering my questions and doublechecking my work.*

*I should also take the time to thank Ed Jaoudi, the electrical technician, and Grant Smith, our machinist, for their advice and expertise.*

*Finally, my thanks go out to the Swarthmore College Department of Engineering in General for the numerous opportunities that they have offered, not just in the course of this project, but throughout my time here.*

## i. Motivation

*One of the most interesting and important topics that I have come across in my science education in general is the study of dielectric waveguides. This subject first came up with the introduction of Maxwell's Equations for electromagnetic waves in Physics 8 (E+M), and was studied more deeply in Physics 112 (Intro to Electrodynamics Seminar) and in Physics 115 (Modern and Quantum Optics).*

*This topic, however, has had more important implications in the research conducted with Professor Moscatelli in the Department of Physics and Astronomy. In that experiment, we were working towards using optical waveguides built from SiON to guide light of two separate wavelengths, producing a pattern of evanescent light fields capable of guiding a neutral atom above the surface of the waveguide itself.*

*Finally, it is likely that my future education and research interest will focus on systems related to optics, so I would like to gain more experience in working with high-power lasers and optical instrumentation.*

*Thus, I have decided to pursue a research-oriented E90 project that incorporates some theory on waveguides themselves with a tangible experiment. I will be measuring and attempting to model the thermodynamic properties of selected waveguides. This task involves studying the relevant theory, designing mounts for the optical components and aligning the experiment, and finally, attempting to automate the experiment in Labview, which will help teach me more about computer interfacing. This might seem like a lot to accomplish in one semester, but I will be working with Professor Molter to carve out a reasonably sized project from among these different possibilities.*

# Index

<b>i. Abstract</b>	<b>i</b>
ii. Acknowledgments	ii
iii. Motivation	iii
<b>I. Introduction</b>	<b>1</b>
<b>II. Theory</b>	<b>2</b>
A. Overview	2
B. Introduction: The Wave Equation and the Nature of Light	2
C. Traditional Waveguides	4
D. Leaky Waveguides	10
E. Our Waveguide: Solutions and Models	12
<b>III. Setup</b>	<b>15</b>
A. Overview	15
B. Existing Experiment and Setup	16
C. Experimental Design: The Waveguide Mount	17
D. Calibration and Alignment	22
E. Labview: Programming Documentation	27
<b>IV. Discussion and Conclusions</b>	<b>39</b>
<b>V. References</b>	<b>39</b>
<b>VI. Appendix</b>	<b>A</b>
A. Matlab Programming	A
B. Initial Mount Considerations	C
C. Thermoelectric Coolers: Overview and Summary	D
D. Labview Documentation I: Laser Power Scanner	F
E. Labview Documentation II: Initial Heater Controller	H
F. Labview Documentation III: Temp Controller PIDII	N

## List of Tables and Figures

<b>Fig. 1:</b> <i>A Traditional Slab Waveguide.</i>	5
<b>Fig. 2:</b> <i>Mode Shapes of Analytic Solutions.</i>	7
<b>Fig.3:</b> <i>Transcendental Equations.</i>	9
<b>Fig. 4:</b> <i>The Input-Output Characteristics of a Leaky-Mode Waveguide.</i>	10
<b>Fig. 5:</b> <i>The Fabry-Perot Effect.</i>	11
<b>Fig.6:</b> <i>A Mode of the Leaky Waveguide.</i>	14
<b>Fig.7:</b> <i>The Experimental Setup.</i>	16
<b>Fig. 8:</b> <i>The Existing Mount Setup.</i>	17
<b>Fig. 9:</b> <i>The Existing Waveguide Mount.</i>	18
<b>Fig. 10:</b> <i>The Resistance Heater.</i>	19
<b>Fig.11:</b> <i>Thermoelectric Coolers.</i>	20
<b>Fig.12:</b> <i>The Temperature Sensor.</i>	21
<b>Fig. 13:</b> <i>The Redesigned Mount.</i>	21
<b>Fig. 14:</b> <i>Initial Calibration Data for the Resistance Heater.</i>	23
<b>Fig. 15:</b> <i>Initial calibration Curve for the Thermoelectric Coolers.</i>	23
<b>Fig. 16:</b> <i>Ideal Performance of the Heater.</i>	25
<b>Fig. 17:</b> <i>Cooling Power vs. Temp. Difference for a Thermoelectric Cooler.</i>	26
<b>Fig.18:</b> <i>Ideal Performance of the TEC Modules.</i>	27
<b>Fig. 19:</b> <i>Step Response of the Cooler Circuit.</i>	29
<b>Fig.20:</b> <i>Proportional Control.</i>	30
<b>Fig. 21:</b> <i>Proportional Control: Simulink.</i>	31
<b>Fig.22:</b> <i>First Attempt at Integral Control.</i>	31
<b>Fig. 23:</b> <i>First Attempt at Integral Control: Simulink.</i>	32
<b>Fig.24:</b> <i>PID Control.</i>	32
<b>Fig. 25:</b> <i>PID Control: Simulink.</i>	33
<b>Fig. 26:</b> <i>Simulink Model for Final PID Controller.</i>	33
<b>Fig.27:</b> <i>Theoretical Limits for TECs.</i>	34
<b>Fig. 28:</b> <i>Step Response of the Heater.</i>	35
<b>Fig.29:</b> <i>Controlled Heater Output.</i>	36
<b>Fig. 30:</b> <i>Controlled Heater Output: Simulink.</i>	36
<b>Fig.31:</b> <i>Output with Trigger: Underdamped.</i>	37
<b>Fig.32:</b> <i>Output with Trigger: Overdamped.</i>	38

**I. Introduction:**

The goal of this project was to examine the wavelength, power, and temperature dependence of a set of existing leaky-mode, slab optical waveguides. The transmission of light through these structures was examined both by analytically computing the shape of the mode that propagates through them, and by rebuilding a previous experiment to measure the transmission ratio through these waveguides as the input power and temperature are varied.

Unfortunately, the main laser for the project—a Ti-Sapphire laser—was in the process of being repaired for much of the semester, so no final data could be collected. Therefore, the experimental portion of this project focused on building a suitable physical setup to achieve the ultimate goal, and on producing LabView code to automate this setup to perform the whole experiment.

In summary, this project was an effort to investigate the nonlinear behavior of leaky-mode waveguides both analytically and experimentally. The following report is an overview of the theory behind this work, as well as a formal documentation of the steps taken to rebuild the automated experiment.

## II. Theory:

### A. Overview

The theoretical portion of this project amounted to solving the differential equations for wave propagation in dielectrics with complicated sets of boundary conditions. This section describes this propagation in general, and then applies this behavior to the leaky case.

The material is presented in the following order: first, the general theory behind electromagnetic waves is included as an introduction. A single-slab waveguide is solved as an example problem. Next, the concept of a leaky waveguide is defined, and the relevant behavior of this system is briefly explained. Finally, the leaky-mode problem is presented; the resulting modes are evaluated and plotted, as an exercise, and in hopes that these solutions might later be generalized to include their temperature-, power-, and wavelength-dependence.

### B. Introduction: The Wave Equation and the Nature of Light<sup>1</sup>

*Note: this section is meant to serve as a brief introduction to light waves in general. The reader is invited to skip this section if he or she has a modest understanding of this material.*

In the classical sense, a ray of light is a well-defined transverse electromagnetic wave. That is to say, it is a solution to the general equation for a traveling, transverse wave:

$$y = f(x - vt). \quad (\text{B.1})$$

Thus  $f$  travels along in space along the  $x$  direction with some propagation velocity,  $v$ . In general, the propagation velocity and the shape of the wave can be anything we might like it to be (a wave traveling along the string of a guitar, for instance, would have a much different velocity than light propagating in a vacuum). We can differentiate  $y$  twice by both  $x$  and  $t$ , introducing the variable  $x' = (x - vt)$ . Assuming that  $f$  is a well-behaved function, we find that  $\frac{\partial^2 y}{\partial x^2} = \frac{\partial^2 f}{\partial x'^2}$  and that  $\frac{\partial^2 y}{\partial t^2} = v^2 \frac{\partial^2 f}{\partial x'^2}$ . Comparing these results, we come up with the well-known, one dimensional, differential wave equation:

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 y}{\partial t^2}. \quad (\text{B.2})$$

We thus have an expression for a one dimensional, transverse wave in the most general form.

---

<sup>1</sup> The Derivation Presented in section B follows from Chapter 8 of Pedrotti & Pedrotti. Introduction to Optics. (Second Edition, Prentice Hall, NJ 1993).

Over the years, however, numerous clever experiments have revealed a substantial amount of information about the actual nature of the light wave. Light is periodic (this can most easily be shown by diffraction, in which the light interferes constructively whenever the path-length difference between two incoming beams is exactly one wavelength). That is to say, we must restrict our wave from some arbitrary shape in space to the set of sinusoidal functions:

$$y = \sin(x - vt)$$

We can also restrict the values in the argument of the sinusoidal function. We know from numerous experiments that light has a well-defined speed,  $c$ ; in fact, relativity is founded on the principle that it would take infinite energy for a particle with mass to achieve this speed, and hence,  $v \rightarrow c$  for light. We also know that monochromatic light has a well-defined wavelength (a particular wavelength of light will give it a unique color). Thus, a ray of monochromatic light will have a well-defined propagation constant  $k$  such that

$$y = \sin[k(x - vt)]$$

which defines the wavelength as  $\lambda \equiv \frac{2\pi}{k}$ . Relating this to the velocity of the wave, which we know is equal to  $c$ , we can easily show that successive peaks will pass a point with frequency:

$$v = c\lambda .$$

The wave can now be rewritten as:

$$y = \sin(kx - \omega t) \tag{B.3}$$

where  $\omega = 2\pi v$  is its angular frequency.

We can now make several extensions to this form. We know the Euler identity for a complex exponential ( $e^{j\theta} = \cos \theta + j \sin \theta$ ), and the fact that in quantum mechanics, a photon is treated as a particle with spin. We can thus see—with the benefit of hindsight—that the wave can be described more aptly as a function that rotates in the real-complex plane (much like a quantum-mechanical wavefunction):

$$y = Ae^{i(kx - \omega t)} \tag{B.4}$$

where  $A$  is the amplitude of the wave. The physically observable quantity would be related to the real component of the wave, squared, as the intensity of this wave is given by  $I = |y|^2 = y^* \cdot y$ .

Our last adjustment will be extending the particle in three dimensions, which is straightforward, because—as the wave always travels along a line—we always know the direction of propagation  $\vec{k}$ . If the wave is at some arbitrary radius  $\vec{r}$ , we know that the distance the wave traveled *along the direction of propagation* is given by  $\vec{k} \cdot \vec{r}$ , and thus, we can write the expression for the wave in three dimensions and time as:

$$y(x, y, z, t) = Ae^{i(\vec{k} \cdot \vec{r} - \omega t)} \quad (\text{B.5})$$

We can immediately plug the expression for an electromagnetic wave into this form. We assume that the transverse-electric component of the wave has the amplitude  $\vec{E}_o$ , which is a vector quantity because it can rest anywhere in the plane normal to the direction of propagation. The expression for the transverse electric field produced by this wave is therefore given by:

$$\vec{E} = \vec{E}_o e^{i(\vec{k} \cdot \vec{r} - \omega t)} \quad (\text{B.6})$$

We know from Maxwell's equations, which govern classical electrodynamics, that this oscillating electric field produces an oscillating magnetic field that is perpendicular to this field, and still perpendicular to the direction of propagation. This field will be in-phase with the electric field:

$$\vec{B} = \vec{B}_o e^{i(\vec{k} \cdot \vec{r} - \omega t)} \quad (\text{B.7})$$

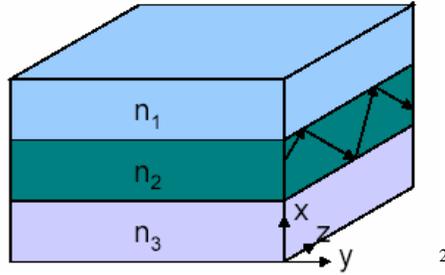
with the relative magnitudes and directions given by:

$$\begin{aligned} E_o &= cB_o \\ \vec{B} \bullet \vec{E} &= 0 \end{aligned} \quad (\text{B.8})$$

Our expressions for the Transverse Electric (TE) and Transverse Magnetic (TM) fields are given in (B.6) and (B.7), with the relative direction and magnitudes given by (B.8).

### C. Traditional Waveguides

A traditional dielectric waveguide directs and confines light within a medium of high refractive index,  $n_2$ , surrounded by media of lower refractive indices,  $n_1$ ,  $n_3$ . An example of a simple waveguide is the slab structure shown in Fig. 1. The light is confined in one dimension (labeled the x axis), by total internal reflection occurring at the junctions between the guiding and cladding layers.



**Fig. 1:** A Traditional Slab Waveguide. The guiding layer has the largest refractive index,  $n_2$ , and thus this medium internally reflects light waves passing through it, confining it in the  $x$ -dimension. The layers above and below the guiding layer are assumed to be infinite in  $x$ , and in this dimension they produce an exponentially decaying wave function; this is known as an evanescent field. We assume that the entire structure is constant and infinite in the  $y$ -dimension, so that the analytic solution is uniform with respect to  $y$ .

We will fully solve this problem below as an example, because its geometry is very similar to the final system in question. The piecewise compilation of the final solution, and the application of boundary conditions to solve for the final coefficients, is a process that we will mirror in our full solution.

To begin our solution, we must start with Maxwell's Equations, which govern the propagation of electromagnetic waves in classical mechanics. They are summarized below:

$$\begin{aligned}\nabla \times \bar{E} &= -\frac{\partial \bar{B}}{\partial t} \\ \nabla \times \bar{H} &= \frac{\partial \bar{D}}{\partial t} + \bar{J} \\ \nabla \cdot \bar{B} &= 0 \\ \nabla \cdot \bar{D} &= \rho\end{aligned}\tag{C.1}$$

We can restrict our medium as linear, isotropic, and nonconductive (this is just an approximation), so that the electric displacement and magnetic induction are linearly related to the electric and magnetic field strengths:

$$\begin{aligned}\bar{D} &= \epsilon \bar{E} \\ \bar{B} &= \mu \bar{H}\end{aligned}\tag{C.2}$$

---

2. Arrand, Helena Francis. Optical Waveguides and Components Based on Porous Silicon, University of Nottingham, September 1997. Pp 18.

where  $\varepsilon$  is the electric permittivity of the material, and  $\mu$  is its permeability. We also know that a nonconductive material can not carry current, so the current density,  $\bar{J}$ , is equal to zero.

We can assume that electromagnetic waves have a periodic time dependence given by:

$$A(t) = Ae^{j\omega_0 t} \quad (C.3)$$

where  $\omega_0$  is the angular frequency of the light. In free space, the relationship between this angular velocity and the wavelength is given by  $\frac{\omega_0}{c} = \frac{2\pi}{\lambda} = k_0$ . The point is that since the light is monochromatic at a controlled frequency coming out of our laser, we know the angular velocity for our system.

If we plug equations (2) and (3) back into Maxwell's equations (1) for the general case, and also apply  $\bar{J} = 0$  for our system, then the Maxwell equations will simplify to:

$$\begin{aligned} \nabla \times \bar{E} &= -j\omega_0 (\mu \bar{H}) \\ \nabla \times \bar{H} &= j\omega_0 (\varepsilon \bar{E}) \end{aligned} \quad (C.4)$$

These equations have an infinite number of solutions, but the loss-free one is a complex exponential in  $\hat{z}$ ; otherwise, light would be lost along the propagation direction, and we would not have total internal reflection. We assume that the light propagates in the  $\hat{z}$ -direction:

$$A(x, y, z) = A(x, y)e^{-j\beta z} \quad (C.5)$$

where  $\beta$  —the propagation constant of the solution—is the component of  $\bar{k}$  in the  $\hat{z}$  direction (see equation (B.5).) Note that the above relation is just one of many possible solutions (more than one propagation constant may be able to solve the boundary conditions. In some cases, none will.)

We can imagine what happens under the circumstances of total internal reflection. To simplify matters, let us assume that the wave has no y-dependence: it travels along the plane of the cross section shown on the right hand side of Fig.1, bouncing back and forth between the interfaces of the guiding and cladding layers, in the x-z plane.

Let us look at what happens immediately before and after the reflection. Equation (C.5) still describes the motion of the wave in the  $\hat{z}$  direction, but we need a similar exponential relation to describe the motion in the x-coordinate. Plugging this in, we get:

$$A(x, y, z) = A(y)e^{-j\beta z} e^{-jk_x x} \quad (C.6)$$

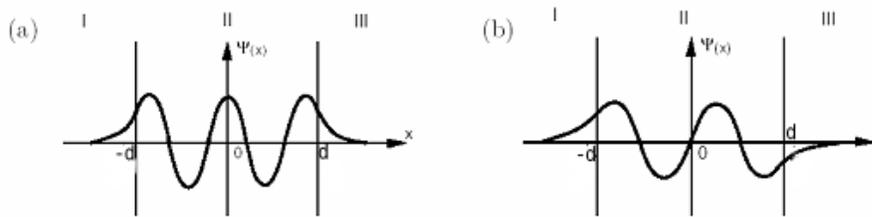
where  $k_x$  is the component of  $\bar{k}$  in the  $\hat{x}$  direction. Immediately after the reflection, however, it changes the x-component of its direction, as it begins traveling in the  $-x$  direction, back towards the center of the guiding layer. Thus,  $k_x \rightarrow -k_x$ , and the second exponent in (C.6) becomes positive. The amplitude and propagation constant remain the same, yielding:

$$A_r(x, y, z) = A_r(y)e^{-j\beta z}e^{+jk_x x}. \quad (C.7)$$

We can add the contributions just before and after the reflection, using the approximation that, as  $z$  has not changed much, the first exponential in each (C.6) and (C.7) will be roughly equal:

$$\begin{aligned} A(x, y, z) &= e^{-j\beta z} (A(y)e^{-jk_x x} + A_r(y)e^{+jk_x x}) \\ &= e^{-j\beta z} [(A + A_r) \cos(k_x x) - j(A - A_r) \sin(k_x x)] \end{aligned} \quad (C.8)$$

The cosine term above is known as the even mode solution, and the sin term is known as the odd mode. Both are allowed to exist, but we can separate our solution into even and odd modes, as each instance will produce different values for  $k_x$  to meet the boundary conditions.



**Fig. 2:** Mode Shapes of Analytic Solutions. An even mode (a) and odd mode (b) solution to the single-slab waveguide.<sup>3</sup>

In order to complete our solution, we must now find the form of the solution outside the guiding layer. We assume that these regions extend to infinity, so to have a physically realizable solution (one in which the total amplitude is integrable to a non-infinite constant), we must have decaying, real exponential solution from (C.4). We let this real coefficient of exponential decay be termed  $\alpha_x$ , and we obtain the following general form:

$$A(x, y, z) = A_1 e^{\alpha_x x} e^{-j\beta z} + A_2 e^{-\alpha_x x} e^{-j\beta z}. \quad (C.9)$$

<sup>3</sup> Green, Mike. "Solutions to the TISE in One Dimension"  
[http://www.ph.rhul.ac.uk/course\\_materials/PH221/Notes/chapter3.pdf](http://www.ph.rhul.ac.uk/course_materials/PH221/Notes/chapter3.pdf)

The coefficients will be determined when we plug in boundary conditions. We note immediately that only  $A_1$  will survive below the boundary layer, and only  $A_2$  above it, in order to keep the solution physically realizable (otherwise the solution would blow up at infinite  $x$ ).

We know now that we can piece together a solution for the whole system using the solutions from different regions, and keeping  $E$  and  $H$  continuous; a discontinuity in either of these fields would require a source of charge or current. We will proceed with a solution for even modes only (note that the solution for odd modes just involves plugging the sine wave into the solution instead of the cosine function).

The task of keeping  $E$  continuous amounts to keeping the piecewise solution for the amplitude of the transverse electric field continuous at the boundary layers. If we set the interfaces between boundary and cladding layers to be at  $\pm d$ , then this boundary condition produces:

$$(A + A_r) \cos(k_x d) = A_2 e^{-\alpha_x d} = A_1 e^{-\alpha_x d} \quad (\text{C.10})$$

So clearly,  $A_2 = A_1$ .

We can also make sure that the continuity of  $H$  is preserved. From the top line of (C.4), we can see that this amounts to making sure that the derivative of  $E$  in each coordinate is continuous. We find the derivatives of the pieces at the interfaces and equate:

$$-k_x (A + A_r) \sin(k_x d) = -\alpha_x A_2 e^{-\alpha_x d}. \quad (\text{C.11})$$

Dividing (C.11) by (C.10), we can eliminate the amplitude coefficients to yield the following transcendental equation:

$$\tan(k_x d) = \frac{\alpha_x d}{k_x d}. \quad (\text{C.12})$$

As a final step, we must find an expression relating  $\alpha_x$  and  $k_x$ . To do so, we derive the dispersion relations: we know that the propagation constant  $\bar{k}$  for a light wave in free space is given by  $|\bar{k}| = \frac{\omega}{c}$  (see (B.3)) and the speed of light in a dielectric material is

given by  $c = \frac{1}{\sqrt{\mu\epsilon}}$ , we know that:

$$|\bar{k}| = \omega \sqrt{\mu\epsilon} \quad (\text{C.13})$$

Finally, we know that, if the light is propagating in the  $x$ - $z$  plane, then the norm, squared propagation vector is equal to the sum of the squares of its  $x$  and  $z$  components: inside the

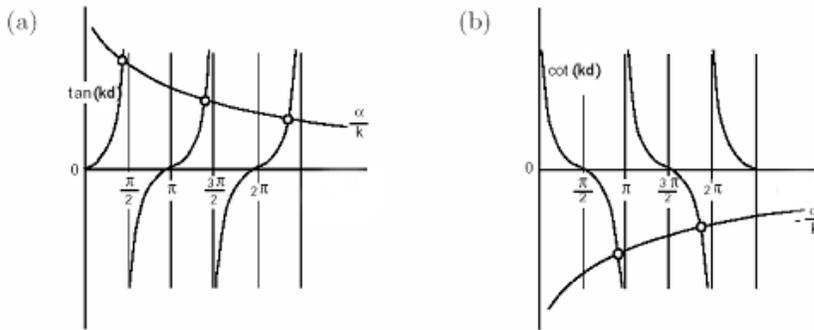
guiding layer,  $|\bar{k}|^2 = k_x^2 + \beta^2$ , and outside,  $|\bar{k}|^2 = -\alpha_x^2 + \beta^2$ . Plugging these relations into (C.13), we get the equations:

$$\begin{aligned}\omega^2 \mu_0 \varepsilon_{in} &= k_x^2 + \beta^2 \\ \omega^2 \mu_0 \varepsilon_{out} &= -\alpha_x^2 + \beta^2\end{aligned}\quad (C.14)$$

Combining these equations, we can get a relation for  $\alpha_x$  in terms of  $k_x$  and fundamental properties of the system:

$$\alpha_x^2 = \omega^2 \mu_0 (\varepsilon_{in} - \varepsilon_{out}) - k_x^2. \quad (C.15)$$

We can use the above relation (C.15), in conjunction with the transcendental equation (C.12) to solve for the even mode solutions. Generally, this is done graphically or numerically: both relations will produce a set of curves, and the modal solutions will occur where these curves intersect.



**Fig.3: Transcendental Equations.** We note that the mode solutions are the intersections of the two functions expressed in the transcendental equation. We show graphs for the even (a) and odd (b) mode solutions.

Since the tangent function passes through the origin in  $k_x - \alpha_x$  space and goes continuously to  $\alpha_x = \infty$ , and equation (C.15) resides in the first quadrant with strictly negative slope, there will always be at least one point of intersection between (C.15) and (C.12). Multimode solutions occur when (C.15) intersects multiple periods of the tangent function in the transcendental equation.

As for the odd mode solutions, the dispersion relation remains the same, but the transcendental equation will be different (sin instead of cosine). This equation then becomes:

<sup>4</sup> Green, Mike. "Solutions to the TISE in One Dimension"  
[http://www.ph.rhul.ac.uk/course\\_materials/PH221/Notes/chapter3.pdf](http://www.ph.rhul.ac.uk/course_materials/PH221/Notes/chapter3.pdf)

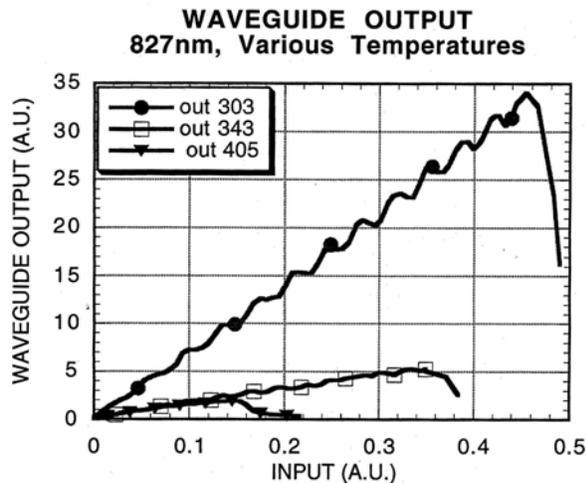
$$-\cot(k_x d) = \frac{\alpha_x d}{k_x d} \quad (\text{C.16})$$

Though multimode behavior can occur with odd modes, the cotangent function does not intersect the origin, and thus, there is no restriction that every system includes at least one odd mode.

#### D. Leaky Waveguides

A leaky waveguide is one in which the guiding layer does not have the highest index of refraction. The solution in this layer does leak light, much like the cladding layers in the single-slab case, and thus, this system is not an efficient method for long optical transmission.

This transmission ratio, however, is sensitive to changes in temperature, wavelength, and input power. The temperature- and power-dependent behavior can be seen in the following graph, which was taken by Cameron Geddes in a previous experiment; these results are the primary motivation for our own project.



**Fig. 4:** *The Input-Output Characteristics of a Leaky-Mode Waveguide.* The transmission ratio is seen to have a periodic dependence on input power, as well as a general dependence on temperature. We see that the transmission ratio is very low for high temperatures.

We can see curves at three distinct temperatures. While the general transmission ratio (slope of the output versus input) is different for each temperature, we can also see a periodic dependence of the absorption ratio on the input power.

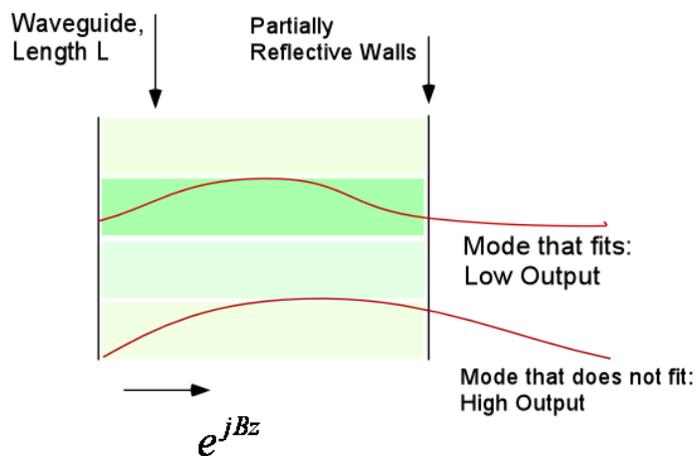
<sup>5</sup> Cameron Geddes, 1993.

We attribute this dependence to the Fabry-Perot Effect, whereby the light reflects from both faces of the waveguides, and, depending on the propagation constant and the overall length of the slab, these reflections will produce an output that interacts either constructively or destructively inside the resonator.

This relationship is periodic because an increased input will heat the slab somewhat, thereby changing the refractive index of the slabs, and ultimately changing the propagation constant for the resulting modes. Since the wavelength of the mode in the  $\hat{z}$  direction is given by

$$\lambda = \frac{2\pi}{\beta^2}, \quad (4.1)$$

the apparent length of the cavity will change, and hence, the mode will go back and forth between one that fits the cavity (i.e. the cavity length is an integer multiple of wavelengths) and modes that do not. We can see this behavior in Fig. 5 below.



**Fig. 5:** *The Fabry-Perot Effect.* The mode propagates in the resonant cavity with a periodic dependence. If this dependence forms a mode that “fits” the resonant cavity, then most of the light will be reflected back into it (top). However, when the mode does not fit, more of the light will escape (bottom). It is believed that this effect is responsible for the periodic dependence seen in Fig. 4.

The amplitude of this periodic effect decreases with temperature, as does the overall transmission ratio. In fact, at higher temperatures, not much light gets through the waveguide at all, so instead, we would like to look at what happens for lower temperatures. It is clear that much of the interesting behavior resides in this area that we

have not yet had the capability to explore, and so it was the goal of this project to produce an experiment that could test the slabs below room temperature.

#### E. Our Waveguide: Solution and Models

Our particular waveguide setup is shown below. It is similar to the one that was used to produce the graph in Fig. 4; the wavefunctions will be determined analytically, in hopes that these results may help uncover the root of the nonlinearities described in Section D.

We begin our solution much like we did in the single-slab case in Section C. Many of the steps are repeated, and the algebra gets rather tedious for this case, so some of the steps will be summarized instead of explicitly computed.

Our setup is shown above. The layer  $n_1$  has the highest index of refraction, so this region has a sinusoidal solution in  $x$ . The input, however, is coupled into the region with  $n_2$ ; this serves as our leaky, guiding layer.

##### *Setting up the Solution:*

We can divide the solution up into regions of decaying exponentials and sinusoids, much like we did for the single-slab case in Section C. However, this solution is much more tedious, and symmetry can not be used to simplify it as easily. Therefore, many of the algebraic manipulations will be skipped in this section, and only the results will be shown. Though they are more complicated, the boundary conditions are applied with the same methods as in the previous problem.

We start with the piecewise solution:

$$\begin{aligned}
 n_0 : & A e^{-\alpha_0(x-d-h)} \\
 n_3 : & C e^{ik_x(x-d)} + D e^{-ik_x(x-d)} \\
 n_2 : & E e^{\alpha_2 x} + F e^{-\alpha_1 x} \\
 n_1 : & G e^{\alpha_1 x}
 \end{aligned} \tag{E.1}$$

We apply the continuity condition for the electric field at the three junctures (we assume that the top and bottom layers extend to infinity):

$$\begin{aligned}
 x = d + h : & \quad A = C e^{ik_x(h)} + D e^{-ik_x(h)} \\
 x = d : & \quad C + D = E e^{\alpha_2 d} + F e^{-\alpha_2 d} \\
 x = 0 : & \quad E + F = G .
 \end{aligned} \tag{E.2}$$

We can also apply the conditions for continuous H (remember that this condition requires a continuity of the x-derivative of the transverse electric field):

$$\begin{aligned}
 x = d + h : \quad & -\alpha_0 A = jk_x (C e^{jk_x(h)} - D e^{-jk_x(h)}) \\
 x = d : \quad & jk_x (C - D) = \alpha_2 (E e^{\alpha_2 d} - F e^{-\alpha_2 d}) \\
 x = 0 : \quad & \alpha_2 (E - F) = \alpha_1 G
 \end{aligned} \tag{E.3}$$

We combine these equations to eliminate each of the coefficients A-G, and come up with a final transcendental equation:

$$2 = e^{-\alpha_2 d} \left[ \frac{\alpha_1 + ik_x}{ik_x(\alpha_1 + \alpha_2)} \right] \cdot \left[ (\alpha_2 + ik_x) + (\alpha_2 - ik_x) e^{i2k_x(h)} e^{i2\tan^{-1}(k_x/\alpha_0)} \right] \tag{E.4}$$

The dispersion relations have the same form as in (C), but there are more layers to include:

$$\begin{aligned}
 \omega^2 \mu_0 \epsilon_0 &= -\alpha_0^2 + \beta^2 \\
 \omega^2 \mu_0 \epsilon_0 n_3^2 &= k_x^2 + \beta^2 \\
 \omega^2 \mu_0 \epsilon_0 n_2^2 &= -\alpha_2^2 + \beta^2 \\
 \omega^2 \mu_0 \epsilon_0 n_1^2 &= -\alpha_1^2 + \beta^2
 \end{aligned} \tag{E.5}$$

We can use the dispersion relationships to find  $\alpha_0$ ,  $\alpha_1$ , and  $\alpha_2$  in terms of  $k_x$ . We can then plug in to make our transcendental equation (E.4) just a function of  $k_x$  and constants of our system. We can solve this equation graphically for  $k_x$ , just like we did in the single-slab case.

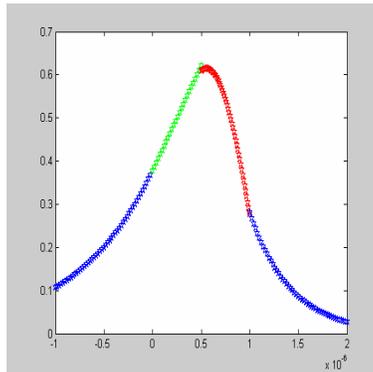
This equation was solved graphically with the help of a Matlab script (see Appendix). This script created an array of acceptable values for  $k_x$ , plotted the transcendental function for each value, and graphed the results with the constant value 2 on the left hand side of the equation for comparison.

#### *Modeling the Solution: Plotting the Modes.*

We can plug the resulting values back into  $\alpha_0$ ,  $\alpha_1$ , and  $\alpha_2$ . If we assume a value for one of the input coefficients—we assume that  $E=1$ —we can use the results in (E.2) and (E.3) to find the rest of the coefficients A-G.

We can now plot the above solutions. This task is again done in Matlab: first the solutions for the coefficients are found by plugging in a solution value from the  $k_x$  array that was plotted earlier. Once we have the coefficients for this particular mode, we can

plot each piecewise solution for an  $x$  array that we create. The results should show a continuous wavefunction with a sinusoidal solution in the  $n_3$  layer, and exponential solutions everywhere else:



**Fig.6:** *A Mode of the Leaky Waveguide.* We can see that the graph is continuous, with a continuous derivative. The one exception is a slight kink in the middle; this error could be removed by improving the accuracy of our numerical solution, and replotting.

#### *Extending the Solution to Include Nonlinearities:*

We now have an analytic solution to the leaky waveguide structure, so we know the values of the coefficients  $k_x$  and  $\alpha_0 - \alpha_2$ . We have also plotted the solution (above) and shown that it fits our continuity criteria for an acceptable solution. While this has been an interesting exercise, it has only served to show that we can solve for our system at a particular temperature. Instead, we are trying to extend our results to include temperature dependent effects.

In order to accomplish this next subtask, a more efficient method of finding the mode solutions would need to be created. Instead of solving the transcendental equation graphically, a search routine would need to be developed to automatically find solutions. This search routine would need to be run for an array of temperatures (and corresponding values for the refractive index of each layer) to produce temperature-dependent values for the propagation constant.

We would also need to find some model for the amount energy that the waveguide would absorb if the input power were raised. Adding this result to the steady state temperature of the waveguide, we would be able to find an input-dependent temperature for our waveguide. Plugging in the appropriate values for the propagation constant at these temperatures, we could come up with a theoretical input-output curve for our system which would mimick the graph in Fig.4. We could then analyze this data, to see if it might be able to produce the sinusoidal input-output dependence shown in section D.

### III. Setup:

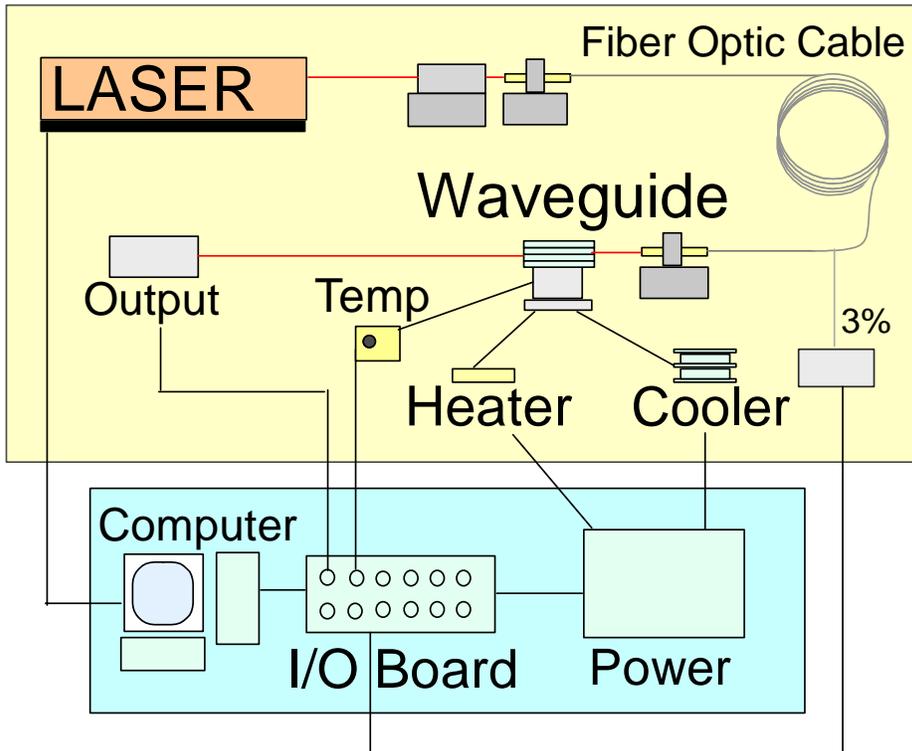
#### A. Overview

The experiment was performed in the Optics and Photonics lab in Hicks basement. The lab includes an optics table with a partial setup and most of the tools that were needed to complete the project. This setup included an Argon Laser that drove a Ti-Sapphire Laser, as well as many of the other components needed (fiber couplers, optical mounts, etc.).

Main adjustments to the existing setup included the addition of a waveguide mount, which was capable of cooling and heating the unit, as well as recording an accurate temperature of the waveguide at any given time. This mount was loosely based on an older version that had been used in the previous rendition of this experiment, but was reconfigured to improve its functionality and accuracy.

Existing LabView code was also updated to work in version 6; additional code was written to control the heating and cooling elements together in the mount and to set the temperature of the waveguide itself.

## B. Existing Experiment and Setup



**Fig.7:** *The Experimental Setup.* The bottom section represents *computer-controlled interfaces*; all of the connections contained in this section are governed by LabView. The *physical setup* is shown in the top section. It includes a LASER, which is coupled into an optical fiber, and then coupled to the waveguide; the output is then collimated and sent to a photodetector. The mount is shown below the waveguide; it includes a heater, cooler, and a temperature sensor, all of which are computer-controlled.

The Laser: The main laser in the setup was an Ti-Sapphire Laser, driven by an Argon Laser. Both lasers were made by Coherent, and both have been in use at the photonics lab for over ten years (the original installation dates back to 1989). Though they had been set up for quite some time, they still needed to be repaired, as some electrical contacts had come loose. They also needed to be fine-tuned for the experiment—a process that involved realigning the lenses and mirrors, which had moved out of place over time (especially retrofitted lenses), as well as cleaning them thoroughly.

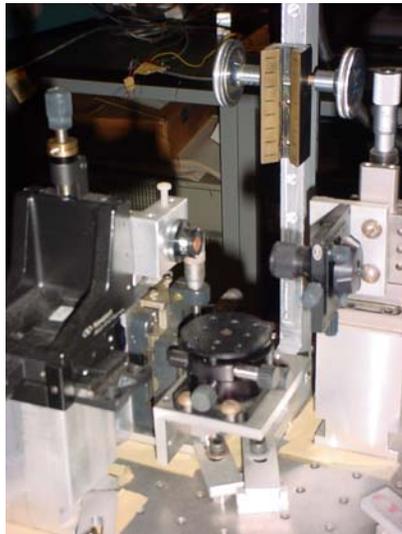
This tuning was done for the most part by Professor Molter, as I had little experience in working with these particular Lasers. It is relevant to mention this step here, however, since the repair of the lasers took a great deal of the semester, so the alignment and cleaning of the laser itself could not start until late April. By the end of the semester, we

had aligned the LASER enough to produce lasing, but we still had not found a proper, single-mode output.

The Fiber Coupler: A fiber coupler existed for the setup, but, much like the laser, this had not been used for quite some time and needed to be repaired and readjusted. The fiber input needed to be re-cleaved and the coupler needed to be realigned, so that the maximum amount of light would couple into the fiber. The realignment procedure, much like most of the work on the laser, was tedious work that needed a more experienced hand, I did learn through careful observation, and I feel that I would have a good idea of how to align the coupler myself, given more time than I had at the moment.

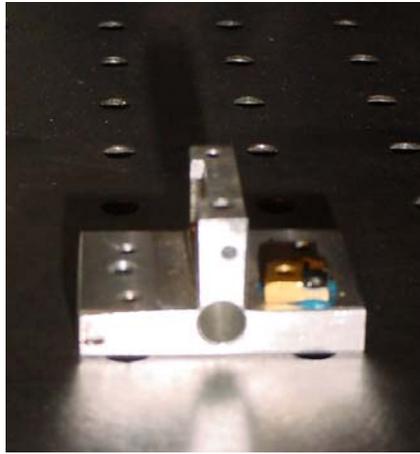
The fiber coupler was already setup to have two outputs: one for measuring the output power (about 3% of the coupled light was split from the original fiber to produce this output), and another (the remaining 97%) to couple into our waveguide.

The Physical Setup: The mount setup consisted of translating (three dimensions of freedom) mounts for the fiber input—on the right hand side of Fig. 8 below—and for a lens to collimate the waveguide output—on the left. At center is a rotating and translating base for the waveguide mount, which was an integral part of the design for this experiment and will be discussed in the following section. Finally, at the rear is a stand for a microscope to align the input through the waveguide itself.



**Fig. 8:** *The Existing Mount Setup.* The mount would be located at the center of the image, where the rotating, translating base sits empty. To its right, the input has a translating mount for a fiber coupler. To the left, we see the objective to collimate the output. A microscope mount sits at the rear.

Existing Waveguide Mount: The existing waveguide mount (see Fig. 9) had the capacity to heat the waveguide through a resistance heater placed in the central hole, and the ability to sense temperature through a silicon diode mounted on the main plate. The cutout on the fin exists to allow the fiber coupler to come arbitrarily close to the waveguide, which would be screwed in on top of this fin.



**Fig. 9:** *The Existing Waveguide Mount.* The brass-colored object on the main plate is the temperature sensor, the hole underneath the fin is to accommodate a resistance heater, and the cutaway on the left (input) side of the fin is to fit a fiber coupler. The waveguide would be mounted on top of the central fin.

This mount was operational, except for a broken temperature sensor, but it did not include all of the functionality that the experiment would need. First of all, data could not be taken below room temperature. Secondly, the temperature sensor was mounted quite far from where the waveguide itself would be—it was considerably closer to the heater than the waveguide would be. This difference in location could lead to inaccurate temperature readings.

The following section will include more details on this design, as well as changes that were made to increase the functionality of the mount.

Circuitry and accessories: The existing setup included most of the wiring and accessories needed to run the project. A computer was provided, with configured links to a DAQ board for Labview, and an appropriate connector to the computer was found. The components of the previous waveguide mount were, for the most part, already wired into the appropriate power supplies or control circuits.

### C. Experimental Design: The Waveguide Mount

The mount for the waveguide was based loosely on a design detailed above. This mount could accommodate the waveguide to fit neatly between a fiber coupler, to provide the input light, and an objective lens, which collimated the outgoing beam. It also fit a

cylindrical resistance heater, which was reused in the current setup; this heater fit just below the central fin, on which the waveguide rested. Finally, the mount included a temperature sensor, which was mounted on top of the main plate.

The new mount used the same general design, but it was extended to include thermoelectric coolers to include cooling capability. Moreover, the temperature sensor was placed much closer to the waveguide, so that temperature readings could be (relatively) accurate. The features of this new mount are detailed below.

The Resistance Heater: This unit was a cylindrical resistor that dissipated power due to heat in a linear fashion ( $P = \frac{V^2}{R}$ ). The resistance of the unit was roughly 25 Ohms, and the maximum voltage was 25 V. Leads were already connected to a computer-controlled power supply.

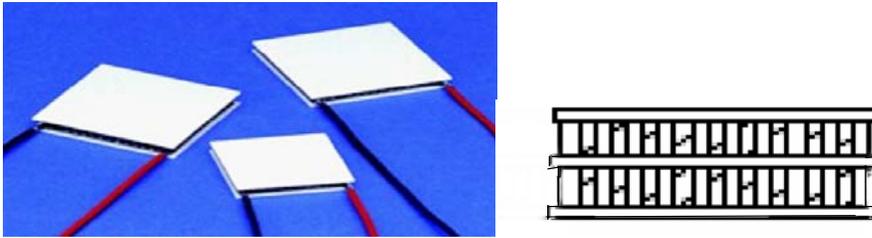


**Fig. 10:** *The Resistance Heater.* It is a cylindrical rod with a resistance cartridge contained within it. Two leads connect to an outside voltage.

#### Thermoelectric Coolers:

We found that the use of thermoelectric coolers (TEC's) was the easiest way to produce accurate, controlled cooling, given our size constraints. When driven by an outside voltage, these units produce a voltage-dependent temperature difference between their top and bottom sides. The cold side was used to remove heat from our mount, while the hot side was heat-sinked in order to lower the temperature of the cold side itself (for a more thorough description of the operation, implementation, and design constraints of TEC's, please see the appendix).

We found that we needed to use a multistage unit, with two TEC's stacked on top of one another, because the ideal cooling range we desired (more than 50K) would be greater than the theoretical limits of one cooler (prepackaged multistage coolers were available through Melcor, but they were expensive, not in stock, and they would require a lead time of at least ten weeks). This stack was selected because it would provide a large amount of cooling, but it would also fit the small space confines of the existing setup.



**Fig.11:** *Thermolectric Coolers.* At left is a picture of some TEC's similar to the ones that were used in our mount.<sup>6</sup> For our setup, a two-stage module was required (right), because the temperature difference sought was greater than the theoretical temperature difference for a single-stage TEC. Two modules were driven in parallel by a computer-controlled source voltage.

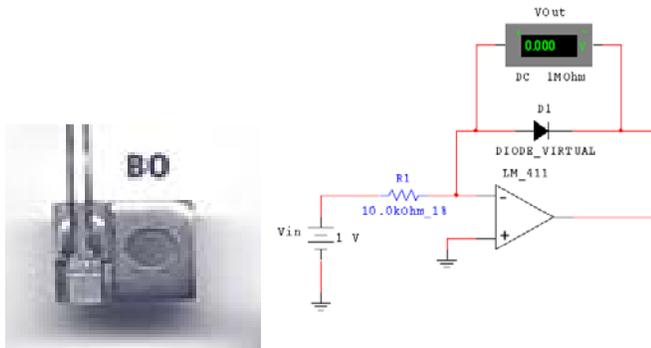
This stack was mounted with thermal grease between the mount and a bonded-fin heatsink. Insulating foam, which was also available from Melcor, was placed around the open edges to help protect the coolers (moisture sealing would be preferable, but it would be expensive, time consuming, and rather permanent). Since the coolers produced a large amount of heat, a fan was needed to distribute airflow to the heatsink. Even slight vibrations on the mount itself could have ruined our alignment, though, so the fan had to be placed on a separate mount alongside the heatsink.

The Temperature Sensor: A new Lake Shore DT-471 BO temperature sensor (shown below) was ordered. This model was the updated version of the one on the previous mount, and it had the same specifications, so it could be used with the same hardware and software setups.

The older temperature sensor was also repaired. This involved filing the existing epoxy from the leads (the unit was moisture-sealed) and reattaching wires to them. This was difficult only because the wires were very fine, and could easily be damaged. This repaired sensor was never used on the mount, but it was incorporated to find the room temperature in the temperature control code (see LabView documentation).

---

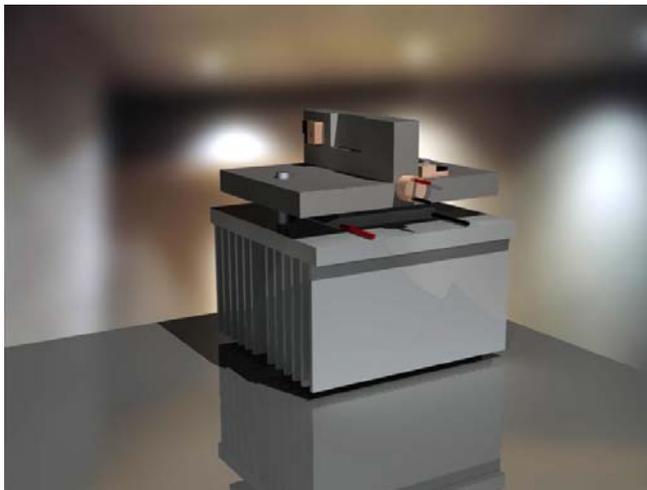
<sup>6</sup> <http://www.melcor.com>



**Fig.12:** *The Temperature Sensor.* It consists of a silicon diode mounted onto a Beryllium Oxide heat sink (left).<sup>7</sup> Ten microamps were provided by the driving circuit shown at the right.

Both temperature sensors needed to be driven by 10 microamps of current to produce accurate temperature readings. Instead of buying an expensive driving unit, two simple Op-Amp circuits were constructed to drive the units (above).

Final Mount Design: The final mount design addressed the previous concerns with the old setup. The mount needed to accommodate the thermoelectric cooler setup detailed above. Secondly, we needed to place the temperature sensor close to the waveguide itself. Finally, the mount had to fit the equipment on the current model, and to fit in the space provided, which was roughly a three-inch square (see Fig. 13). The final design is shown below:



**Fig. 13:** *The Redesigned Mount.* The waveguide itself would sit in the center of the top fin. The two small, brass-colored objects on the top surface are temperature sensors. The cylindrical resistance heater is situated just below the central fin, and the thermoelectric coolers are located between the top plate and the heat sink.

<sup>7</sup> <http://www.lakeshore.com>

The mount was enlarged to accommodate the thermoelectric coolers that were selected, which were approximately one inch, square along its base. This size was needed to ensure that the temperature at the waveguide was even; a smaller cooler would create a larger temperature gradient near the center of the mount, which would be a problem, since the temperature could not be measured exactly at the waveguide due to space limitations. Thus, the mount was expanded, and set to fit a standard, two-inch square heatsink.

Great care was also taken in locating the temperature sensor. While the old mount had the sensor epoxied onto the main plate, it was found that there would be enough room along the main fin—in the new setup—to place the sensor on the input side, very close to the actual waveguide (see Fig. 13). The sensor was also clamped down directly to the mount while it was being fastened, so as to produce a direct thermal contact with the mount without a thick layer of epoxy in between (in the previous design, there was about a millimeter of epoxy between the temperature sensor and the mount itself). These steps would certainly help the sensor provide a much more accurate estimate of the actual temperature at the waveguide.

The mount was designed to meet all of the other specifications of the previous one. The heater was placed in the same location; the cutout on the main fin remained, unaltered; and the thickness of the main plate, as well as the height of the central fin, were matched to ensure that there would be sufficient room to fit the fiber coupler and the collimating lens to be properly aligned.

Finally, the total dimensions of the mount had to fit within the requirements. This became a problem, as we will see later, because the heat sink itself needed to be chopped down considerably in order to fit the mount, two TEC modules, and the waveguide within the given height. This significantly limited the effectiveness of the mount.

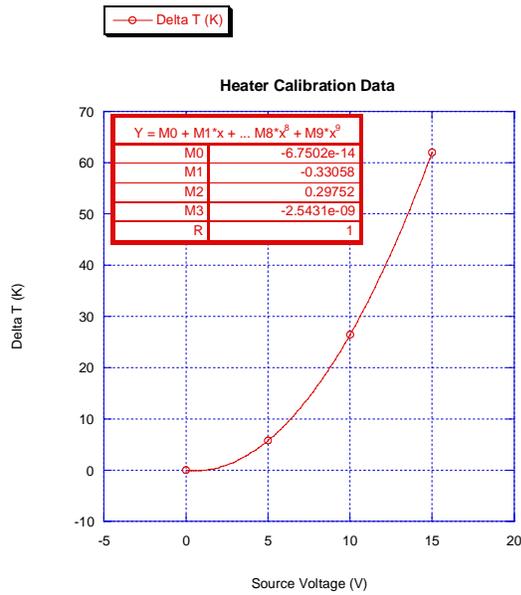
Compounding the problem was the fact that a fan could not be attached to the mount itself, because induced vibrations could ruin the alignment of the experiment. The fan was placed close to the sink, and a shroud was later devised to better direct air into the heatsink. Unfortunately, the main problem was that there was not enough of a thermal contact surface for this convective heat exchange to be very effective.

#### D. Calibration and Alignment

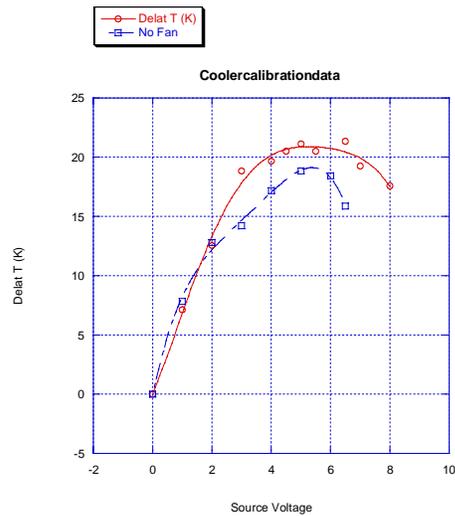
Once the mount was built and the setup was put together, the heater and cooler needed to be aligned. It was desired to know how much temperature change we could achieve for both devices, and it was originally thought that the steady-state Temperature/Voltage curves would be used to set the waveguide to a given temperature (since the experiment was later automated, it turned out that temperature control was achieved through negative feedback, as can be seen in section E).

Preliminary Calibration Curves: Below are the steady-state calibration curves of voltage versus temperature change for both the heater and thermoelectric cooler systems, along

with appropriate polynomial fits to interpolate the data. The heater had a maximum voltage of 25V, while the maximum for the TEC's was 8.6V.



**Fig. 14:** Initial Calibration Data for the Resistance Heater. We can see a polynomial fit at the top left, with a high correlation coefficient.



**Fig. 15:** Initial calibration Curve for the Thermoelectric Coolers. This graph shows temperature versus source voltage with a small 12-V fan on (top) or off (bottom). The fan has surprisingly little effect on the steady state temperature.

From the above data, it is clear that the heating process is much more robust than cooling at the moment. While the heater can easily reach a 50 K temperature change at half its maximum voltage (which is about  $\frac{1}{4}$  of the maximum power), the TECs never achieve more than 25 K.

The reason for this discrepancy is the fact that, with the amount of power that the coolers are using, they produce a large net amount of heat, which can not be properly handled given the spatial constraints of the setup.

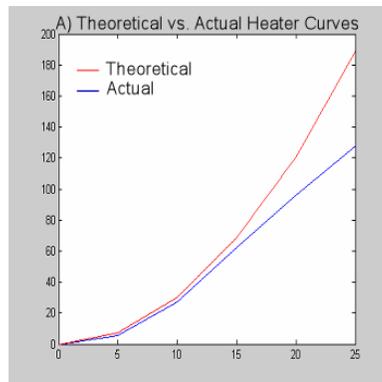
Moreover, the fan had surprisingly little effect. At first this was believed to be because the airflow was not going mainly through the heat sink: it could have been diverted around it. However, putting a shroud around the fan output has had no noticeable effects on the temperature change, which suggests that the main culprit is the heat sink itself.

Theoretical Curves: Ideal curves could be obtained for the heater and cooler if we neglected convective heat transfer and only considered the static temperature load, and assumed that the hot side of the combined TEC would remain at room temperature. We find that the heater will have a parabolic T/V curve, due to the relationship between power and voltage for a resistor, and that the thermoelectric coolers will reach some theoretical temperature difference before they reach the maximum voltage, because efficiency drops off.

Both of these effects match the general shape of the calibration curves above, but we want to go into greater detail to see if we can achieve the specified heating and cooling with our given setup, or whether we would need to redesign the mount itself.

#### *Resistance Heater*

For the resistance heater, we expect parabolic temperature dependence for a constant temperature load. We estimated, in selecting our TECs, that our static load was about 5.26 mW/K. We found the power at each desired voltage, multiplied by this constant of proportionality, and compared to the experimentally obtained results:



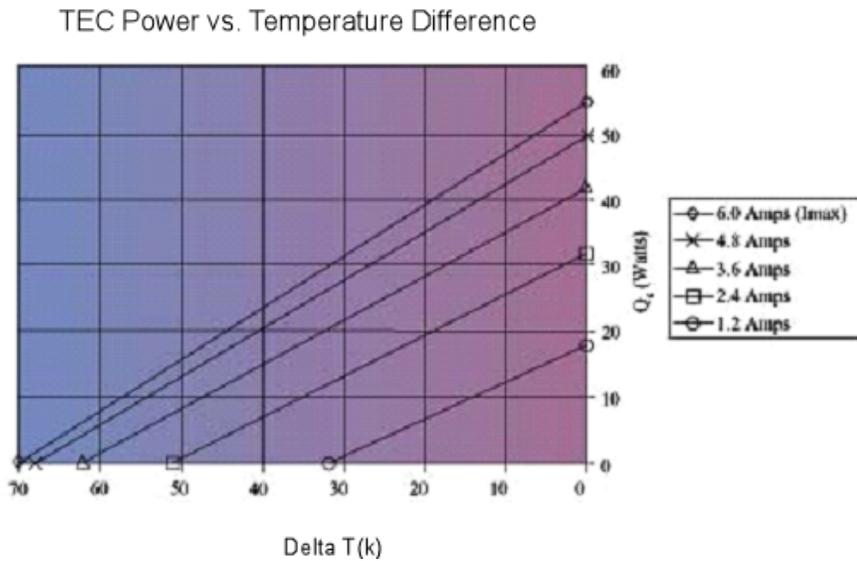
**Fig. 16:** *Ideal Performance of the Heater.* We can see that the actual (bottom) versus ideal (top) performance curves diverge at large temperatures.

The theory fits the experiment well for lower temperatures, but at higher power, convection takes over, and the parabolic curve begins to bend back towards a line. This effect is only prevalent at temperature differences of about 100 K or larger, though, and we are only hoping to heat the setup by 100 K in the first place. This desired heating is achieved with about 22 V, and we could eventually go as high as 120 K for the full 25 W.

#### *TEC's: A Theoretical Maximum Temperature Difference*

At higher levels, all TECs become less effective, because they generate a large amount of heat as waste heat from the power flowing through them, but the effective cooling power begins to level off at higher temperature difference (see Fig. 17 below), as well as near the current maximum. Thus, as further power is applied, the hot side gets much hotter, while the cooler side does not get equally cooler, leading to a net heating effect.

This leads to the existence of a theoretical temperature difference; all TEC modules will reach an equilibrium between the effective amount of cooling they can sustain and the requirements of the system. Driving them beyond this point will cause them to produce more heat without producing any larger temperature difference across them, leading to a net *heating* effect with increasing voltage. This state is unstable and of no use, even though it is within the current/voltage limitations of the TEC module. We will discuss this overdriven state in more depth later (see Section E).



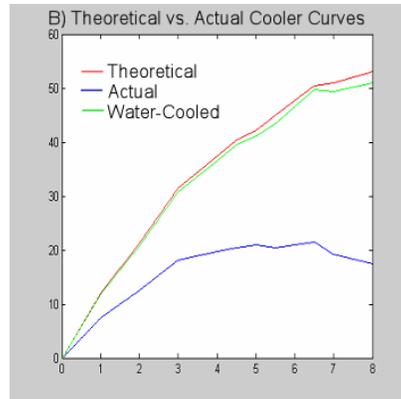
**Fig. 17:** The Cooling Power (vertical axis) vs. Temperature Difference (horizontal) for a Typical Thermoelectric Cooler.<sup>8</sup> It is evident that this cooler can not produce much cooling power above a 60-degree temperature difference which is typical of single-stage TEC's. Since our cooler has two stages, the theoretical temperature limit would be significantly higher, and our unit would be able to deliver significant cooling power at over 50 Kelvin temperature difference, which was our initial goal. Note, however, that the cooling becomes much less effective higher in the power spectrum (each successive line depicts a linear increase in power, but the cooling power becomes less and less sensitive to this increase).

The theoretical temperature difference for a single-stage TEC is at most 40 to 50 K, depending upon the system and the module itself. For temperature differences between 40 and 65 K<sup>9</sup>—which would encompass our present goal—a two-stage TEC module is generally used. This unit would literally consist of two TEC modules stacked on top of one another (see Appendix ), so that the temperature difference is divided between them. Each module can then handle more cooling load, and the total system can reach a lower temperature.

Since we could not afford a prepared multistage array in terms of time or money, we did not have a calibrated specification sheet where we could just read off the properties of our total unit. Instead, the specifications from a single unit were used; these included graphs such as Fig. 17, with results for different hot-side temperatures. It was then assumed that the temperature difference was split evenly between the TEC's, and the hotter of the two was always assumed to operate at room temperature. The equilibrium state of the system was found—given just the static loading case—and plotted below.

<sup>8</sup> <http://www.melcor.com>

<sup>9</sup> This information was provided by a sales representative from Melcor, inc.



**Fig.18:** *Ideal Performance of the TEC Modules.* The actual cooling (bottom) is much lower than what could be achieved ideally (top) or by water-cooling the system (middle).

We notice about a 35 degree increase in possible cooling, which yields about 55 degrees, total. It would not be possible to achieve this full effect, due to convective loading, but since the temperature differences here are much smaller than in the heater, this effect would not be nearly as prominent as it was in the heater.

We can make one final prediction: for water cooled systems, the hot side temperature only rises by about 2-5 degrees. We therefore plot the anticipated fluid-cooled properties above. We can see that, neglecting convective loading, we can get about 52 degrees out of this system, so it is perfectly reasonable to assume that, even with convective loading, we might still be able to obtain at least 45 degrees.

This conclusion has been supported by preliminary tests that were run on the coolers before the mount was installed into the setup. Several trials were run with the fan attached to the bottom of the heat sink—blowing directly onto the hot side heat exchanger—before much of the fin material had been removed to make room in the setup. Temperature differences of about 30-35 degrees were achieved.

If our conclusion is valid, a decision must be made about whether or not it is reasonable to water-cool the system. This option would require a pump and radiator assembly, as well as a water block to attach to the bottom of the TEC. This assembly would wind up costing about a hundred dollars, and it would take significant time to put together (let alone the added hassle of maintaining it).

#### E. Labview: Programming Documentation

The first task to complete in the LabView segment of this project was to recompile and update older versions of code so that they would work under Version 6. Two particularly helpful files were the *Innova Power Scan vi* and the *Temperature Controller IV vi* (see Appendix). The power scan code, which had the laser scan across the full range of power

and write to a file, was repaired by a technician, since the flaw was due to a particularly notorious change in the way VISA (serial) communication was done in newer versions. Furthermore, the original wiring for the laser, and the program itself, were extremely complicated, and it would take an unreasonable amount of time to fix for someone not familiar with that setup.

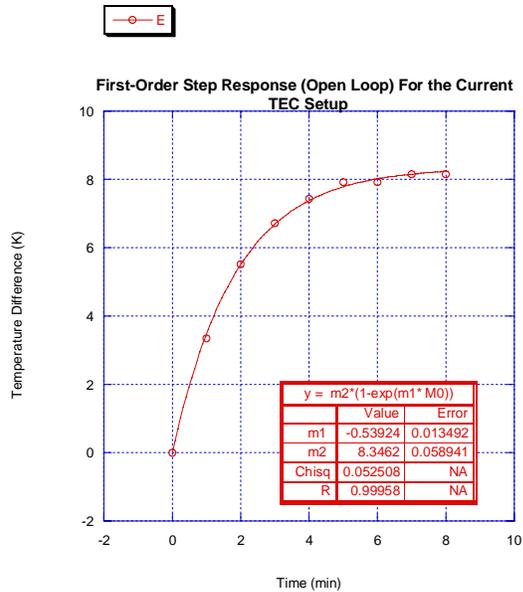
The temperature controller used a simple proportional and derivative controller to adjust to a specific temperature. Once the system was within the target range from the desired temperature, the controller would switch from “seek” mode to two progressively finer settling mechanisms. In essence, this controller worked like a bang-bang controller even when properly configured, and the result was a controller that had many parameters to adjust, but ultimately one that was not as robust as just a simple PID controller, or a phase lead or lag.

This system was overly complicated: the code ran rather slowly, even on modern systems, and the corresponding power supply would be continually switching on and off at high rates of power and at regular intervals just to keep the system within several degrees of the desired temperature. It remains a mystery as to why the authors chose a PD controller, because this drove the response to be unstable, and this controller had no practical application, since the settling time of the system is of minimal importance for our application.

Updated Temperature Controller: The updated temperature controller is moderately simpler than the original code. It uses a PID controller to implement both heating and cooling, and it requires a reasonable set of three coefficients for each process. However, this controller does not rely on several obscure set temperatures, and have numerous modes. The controller functions well for both heating and cooling to produce a well-damped response.

This system needs no “continuous control” mode because it does not wander from the desired temperature. Once it sets to a temperature, the controller remains within a tenth of a degree from the target. Moreover, the response of the controller is tailored to not saturate the power supply, as this may cause overheating and unpredictable results. In summary, this controller produces a more predictable, more stable, and simpler response than the previous “seek” and “acquire” solution. It would thus be easier to use and learn by new students, in addition to being holding a much more consistent temperature reading. The one drawback is that this response might be slower than the other system, but response time is not a major concern in our case.

The code works (as is also detailed in the Appendix) by treating both the heater and coolers as first-order plant transfer functions. As we can see Figs. 19 and 27, the response of both these systems very much resembled a single exponential, except at the high-power limit, when convection and inherent inefficiencies began to severely limit the performance of these systems:



**Fig. 19:** Step Response of the Cooler Circuit. This agrees with a first-order system, so we assume that the transfer function is roughly first-order.

We now model the control function as a closed-loop transfer function with a sample-and-hold function, as temperature could only be taken discretely (at most once every five seconds for our case) in LabView. We will use the result from Phillips and Harbor chapter 13<sup>10</sup> for a PID controller in a discrete system. Our plant transfer function for the cooler is given by:

$$G_c(s) = \frac{0.02}{s + .003}.$$

We solve the transfer function with a twenty-second sample-and hold function (LabView can only take temperature data once per run cycle). We find the z-representation of this plant transfer function. We then find the product of this with a general discrete-time PID controller.<sup>11</sup> Once we plug in appropriate values for the settling time and for the poles of this transfer function, our method yields  $K_d = .35$  and  $z_0 = .9$ . We find our other coefficients from the relations<sup>12</sup>

$$K_p = K_d z_0,$$

$$K_i = \frac{K_d - K_p}{T},$$

<sup>10</sup> Phillips and Harbor. Feedback Control Systems. (Prentice Hall, Saddle River, NJ, 2000), Chapter 13

<sup>11</sup> Phillips and Harbor. p 570.

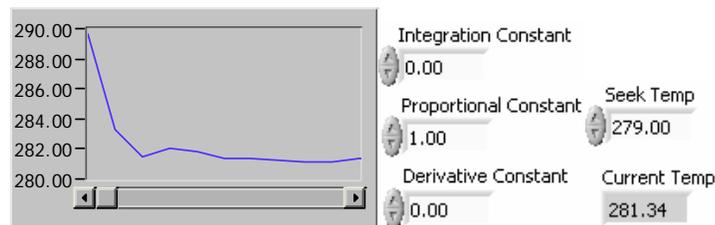
<sup>12</sup> Phillips and Harbor. p 570.

which yield:  $K_p = .31$  and  $K_i = .03$ . These values were later plugged into the PID controller and adjusted slightly (remember that our system is not actually first order, so these values are just a model of how our system might act. We need to make sure that we have a stable response over the whole range of cooling, so we

We note that, given purely derivative control and a maximum temperature difference of about 20 degrees, the step response would saturate the power supply (6V max) at  $K_p = .3$ , so we should be using the full range of the power supply, but we should not be saturating it (at least not for an extended period of time). This hypothesis is confirmed below; if this had not been the case, we would have needed to slow down the response until our expectations were consistent with the physical limits of the system.

#### *First Iteration:*

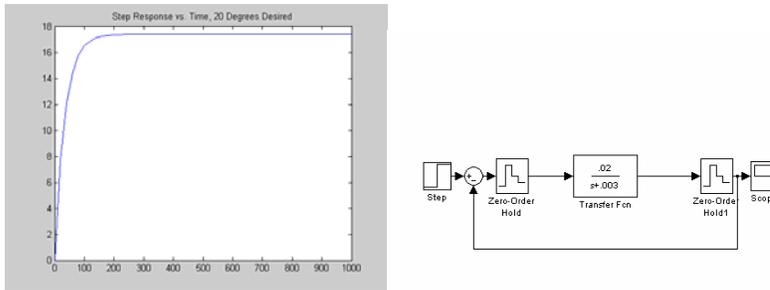
The first iteration of this new code was just to implement proportional control. The response was fairly quick, and there was no stability worry. The code ran simply and smoothly (this would be a good algorithm to start using if the existing PID control stops working). We can see the response plotted below.



**Fig.20: Proportional Control.** Real time temperature output (left) and the relevant coefficients (right) for a proportionally-controlled step response.

The problem with this response, however, is that there was a slight amount of steady state error associated with it: note the difference between the set temperature and the steady state. This could have been corrected by overestimating the temperature difference slightly, but this would not have been interesting from a design standpoint.

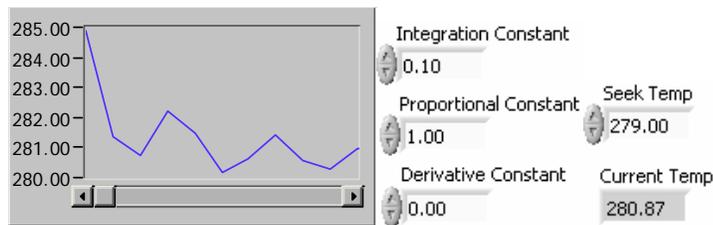
A simulink representation of a similar system is shown below. We use a zero order sample-and-hold function to model the discrete-time measurement process:



**Fig. 21: Proportional Control: Simulink.** Step response of the proportionally controlled cooler circuit in SimuLink (left) and the system model (right). The gain block represents the gain between the voltage output to the power supply, and the voltage that it provided. This response agrees with the real-time data in Fig. 16.

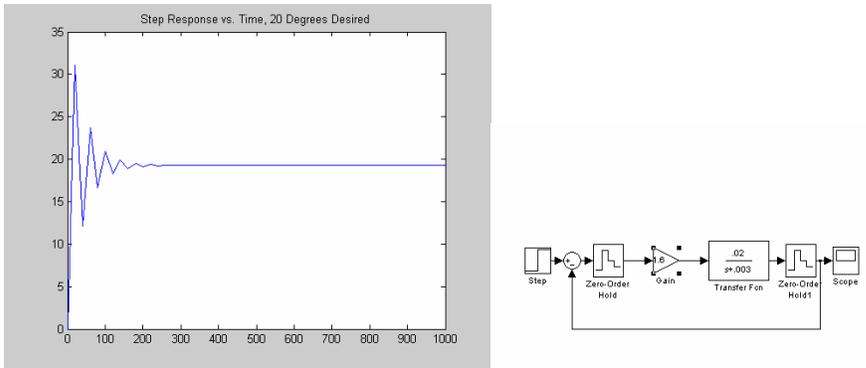
### *Second Iteration: A First shot at Integration.*

While the proportional and derivative controllers were easy to implement in LabView, integral control was not as straightforward, though it turned out not to be as outlandish as was originally feared. The first attempt at integration involved storing an error reading at three (the minimum) times during a run cycle, and then integrating these results over one cycle using LabView's numerical integration subblock.



**Fig.22: First Attempt at Integral Control.** Real time temperature output (left) and the relevant coefficients (right) for a “first-iteration integral”-controlled step response. We can see that this response is fairly unstable.

This approach not only slowed the code down quite a bit because of its complexity, but it also failed to perform a real integration: it just took the integral over one run cycle, thereby increasing the proportional gain. This integration routine caused the response to be more unstable (see below), which agrees with the theoretical assumption, but since there was no functional gain, this method was quickly abandoned.

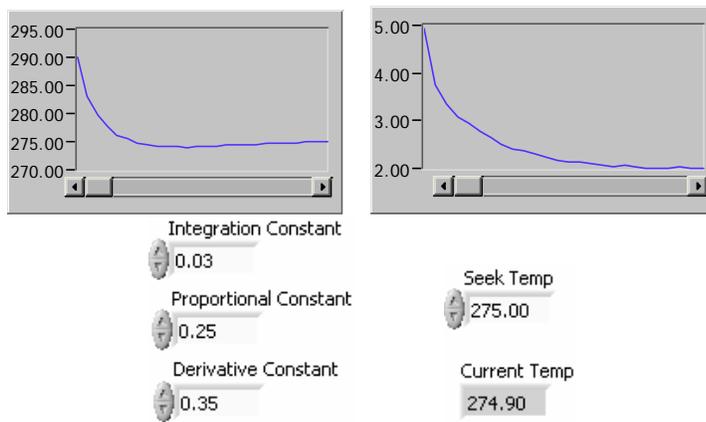


**Fig. 23:** *First Attempt at Integral Control: Simulink.* Step response of the “first iteration derivative” cooler circuit (left) and Simulink model (right). We see that the instability occurs because we have only served to increase the gain of the stable system.

Above is a Simulink model of our expected output (see Fig. 23). The model exhibits the same general behavior as we had in the physical system. An increase of the proportional gain will cause the system to be less stable (here we have increased it to 1.15, instead of 1.10 for the actual controller, to exaggerate the effects).

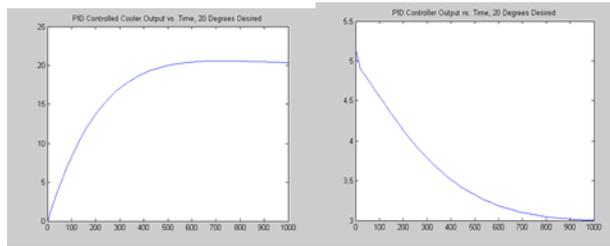
*Third Iteration: Improved Response*

The final integration subroutine used several read/write sub vi’s to write to each other and step the final result up by the amount of error signal each run cycle. This in turn was fed to the rest of the control, and the results were immediately improved. There was less than a degree overshoot, with no ringing: this was the ideal response for the minimum settling time. A controller-voltage graph was added to make sure that the controller, in fact, did not saturate.

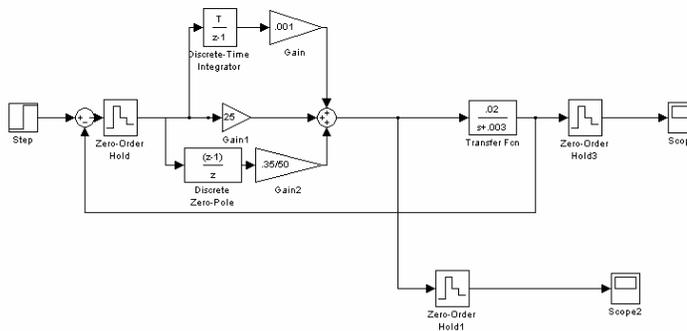


**Fig.24:** *PID Control.* Real time temperature output (left) and the relevant coefficients (right) for a PID-controlled step response.

A simulink printout of the expected response is also shown in Fig. 25. We can see a slight overshoot of less than one degree, which we have above, as well as a cooling graph that resembles the one in Fig. 24.



**Fig. 25:** *PID Control: Simulink.* Step response of the cooler circuit (left), with the controller output (right). This agrees with the real-time data above. We note an overshoot of less than one degree, and a steady-state error of zero.



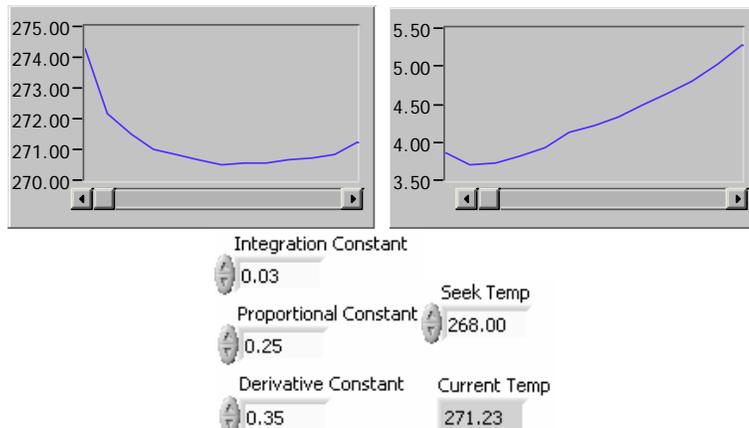
**Fig. 26:** *Simulink Model for Final PID Controller.* The multiple gains shown are due to the gain inherent in the power supply, as well as those contained within the sub-vi's that I used.

We see remarkable agreement between the real and modeled systems. The overshoot values (about 1 degree) and the general shape of the output functions coincide perfectly, and the slight differences in the controller voltage are due to the fact that the system is not perfectly modeled (it is not second order). Furthermore, the sample-and-hold times can be irregular in practice and the derivative function is not exact; rather, the code finds the difference between two points within the subroutine.

#### *Pushing the Limits: How Much Cooling Can We Get?*

Heat buildup in the TEC's has been a problem for the duration of this project. It was determined that above about 5 Volts, the heaters became very inefficient, and they began to produce more heat than cooling. This would lead to an overall decrease in the temperature change, even with an increase of voltage. The nature of this inverse relationship would, in theory, cause a breakdown of the traditional negative-feedback mechanisms on which most control systems rely. In this case, a controller would keep pushing the circuit higher and higher, because a sustained error signal is obtained (this

constant error is then integrated to produce an upwards ramp in the integral controller). As the controller kept pushing, however, the circuit would become more and more feeble, which would eventually result in saturation as the only stable equilibrium. We see this behavior below in Fig. 27.



**Fig.27:** *Theoretical Limits for TECs.* Real-time temperature output (left) and the relevant coefficients (right) for a PID-controlled step response, driven beyond the ideal temperature difference. We can see that the response becomes unstable, with the controller output increasing, but the temperature difference decreasing.

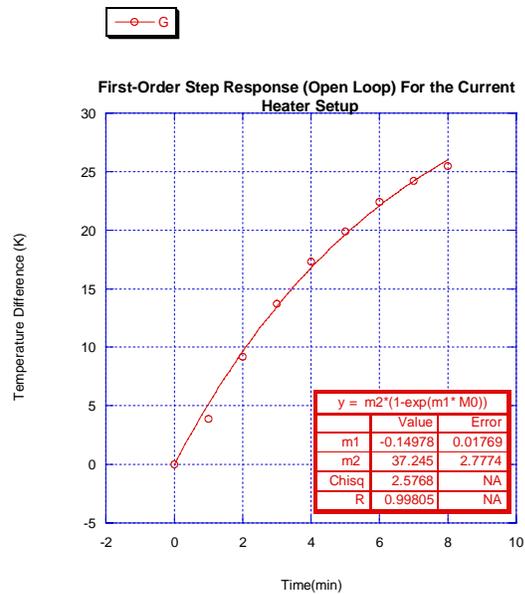
It now becomes clear, at least in the case of the coolers, why avoiding saturation was so important. If we overdrive the controller for a sustained period of time, it will tend to stay saturated rather than to return to a more efficient and reasonable level. Our first order predictions are invalid in this region, so we would need another model to be able to predict these results.

If such a model were built, we could theoretically expand our cooling range to reach the lowest-possible temperature (about 270.5 K from fig. 27). However, since we can easily reach about 272 K with our current model, and because of the much larger effects of the hot-side heat exchanger, our efforts would be best served to find a more efficient heat sink.

#### *Heater Control: A Complete Temperature Controller*

In coming up with the above approach for the cooler, we realize that it would be ideal to incorporate a similar controller for the heater, with different PID constants, and just have a manual or automatic switch that tells the computer which code to use.

We can see the response of the system below in Fig. 28. We assume again that the relationship between input power and voltage is linearly proportional (we use the slope of the secant between 5 V and 20 V in the temperature difference calibration curve).



**Fig. 28: Step Response of the Heater.** Step response of the open-loop heater circuit. This agrees with a first-order system, so we assume that the transfer function is roughly first-order.

With the above results, we can find the transfer function in s-space as:

$$G_h = \frac{.005}{s + .0015}$$

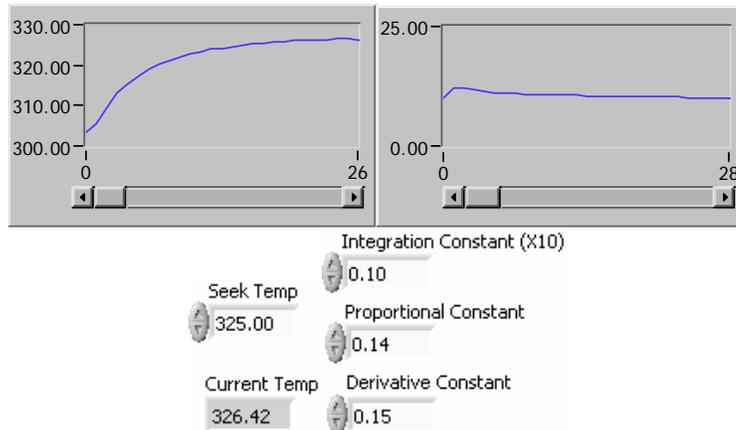
We go through the same rigors as before with the heater, find the PID coefficients:

$$K_p = .14$$

$$K_i = .01$$

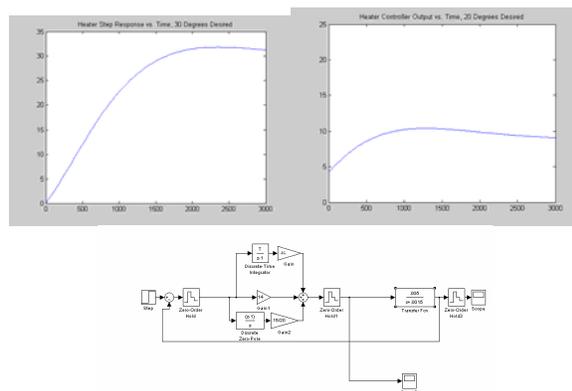
$$K_d = .15$$

and we observe the following behavior:



**Fig.29:** *Controlled Heater Output.* Real time-temperature output (left) and the relevant coefficients (right) for a proportionally-controlled heater step response.

The results are confirmed again by Simulink in Fig. 30.



**Fig. 30:** *Controlled Heater Output: Simulink.* Step response of the heater circuit. The controller is similar to that shown in the diagram of Fig. 29. This response agrees with the ideal one shown in Fig. 25: about one degree of overshoot, and zero steady state error.

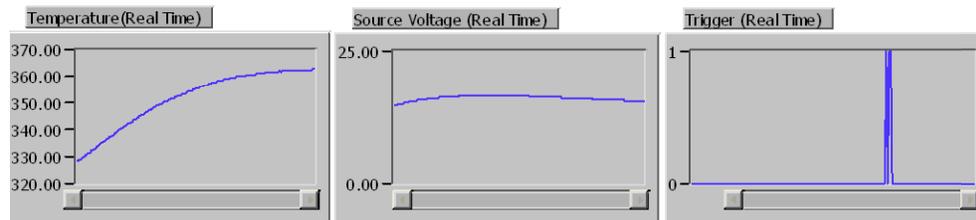
Now, we can use these results, as well as those for the cooler control to come up with a total temperature control unit that will perform both heating and cooling. If we give it an input (T/F switch), we can choose to send the cooling coefficients to the cooling supply, or to send those coefficients used for the heating to the heater supply. This T/F statement can just be generated by a comparison between the set temperature and room temperature, so this code can work seamlessly in an automation code to control temperature throughout the possible range.

This Temperature control code is documented formally in the Appendix.

*A Top-Level File to Integrate the LASER Code and Temperature Controller*

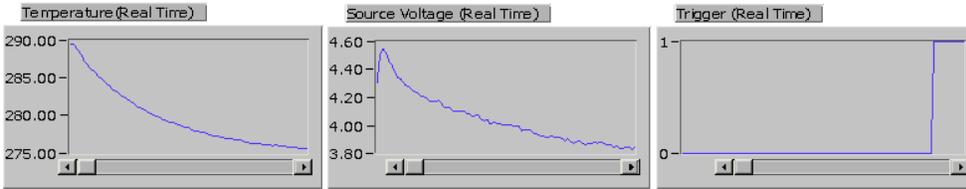
The final LabView Subtask was to write a complete controller to scan temperature and laser output power. Since temperature was by far the slowest response, this code was built to set to a temperature and then to run the LASER code with the trigger response. This laser code was already configured to write the input and output photodiode voltages to a file, and was later modified to include a temperature reading as well.

The first step towards this process was to ensure that the trigger would go off only when steady state was achieved. In previous renditions of the heater or cooler control code, underdamped response was used to produce a faster settling time. However, it was found that a trigger would be much more difficult to design for this process: instead of just setting the trigger to go off when the system got arbitrarily close to steady state, we instead would need to weed out the overshoot and wait for the system to actually come to steady state. Otherwise, the trigger would tend to go off whenever the temperature would be close to the desired temperature, even if the system had not achieved steady state (see Fig. 31).



**Fig.31:** *Output with Trigger: Underdamped.* Real time-temperature output (left), controller output (center), and trigger output (right) for the underdamped controller. We see that the trigger tends to go off when the overshoot occurs.

To solve this problem, PID constants were found for both the heater and the cooler that would provide slightly overdamped responses throughout the range of cooling and heating (this was done by decreasing the integration constant). This response was slower, but it avoided accidental triggering. The trigger was further refined to include a limitation on the derivative; this limit was done to avoid the unlikely event that the sensor would mistake one temperature reading to be within the desired range. We can see a much improved trigger response in Fig. 32.



**Fig.32:** *Output with Trigger: Overdamped.* Real time-temperature output (left), controller output (center), and trigger output (right) for the overdamped controller. We see that the trigger is much more stable than the overdamped case. The time response of the system has slowed down considerably. Instead of setting within two or three minutes, the cooler system takes about five minutes to reach steady state. Instead of setting within ten minutes, the heater takes almost twenty minutes to achieve steady state within one degree of the desired temperature.

Once the trigger was stabilized, the first element of the top-level file was written. This segment was essentially a for-loop which would set to temperatures in user-defined increments. Given a minimum and maximum temperature, and a number of increments, this code would step up from the coldest to the hottest temperature. This configuration was chosen instead of a sequence from hot to cold because of the nature of the setup: the TECs already had a problem with removing waste heat, so going from colder to hotter would only lower their effectiveness.

Within this temperature sensing loop, a second segment of code was written. This second frame would keep controlling the temperature while the LASER scanner ran. The current temperature would be written to a file along with the current fed to the LASER and the input and output voltages from the photodiodes. Thus, we would be able to scan temperature, scan laser voltage, and ensure that our results were accurate (i.e. the temperature didn't wander off, or the experiment didn't go off of alignment).

The wavelength of the LASER would be controlled with a servomotor, but unfortunately, a LASER wavelength scanner could not be incorporated into this module. An old scanning vi was not found, and the LASER was still not well-enough aligned to come up with a new calibration curve. However, once this old code is found, or a new one is written, it could be implemented fairly easily and painlessly between the temperature scanning section of the top-level code and the power scanner.

In summary, a LabView temperature controller was written to seamlessly implement both heating and cooling. A top-level file was then written to scan temperature from low to high, while triggering the laser power scanner. Both modules were demonstrated to work as expected, with the final product being an automated experiment that can measure the temperature and power dependence of the transmission ratio.

#### IV. Discussion and Conclusions:

Unfortunately, it has been impossible to couple the LASER output into the optical fiber, so no final data has been taken. The LASER itself needs to be realigned with the help of a technician, and once a single-mode output would be achieved, the output could be coupled effectively.

The brunt of this project, however, has focused on rebuilding and improving an existing setup. Calibration data has been used to accurately model our revamped system, and this model was later used to produce accurate and efficient control of the heater and cooler, and a top-level file was written to produce an automated process whereby temperature and power can be varied.

We have also shown that analytic solutions do exist for leaky waveguides, and have found the mode solutions at room temperature. The existing theory can then be built upon and expanded to analyze the physical results that our experiment would produce. All that remains to be done is the development of a search routine that can find the temperature dependent output effects of the Fabry-Perot theory and to test these assumptions with our initial data.

As this experiment will continue into the summer, these initial successes can be used to acquire data and to realize the rest of our initial goals, some of which were outside the scope of a one-semester project anyway. Most of the intermediate goals for this project have been met, so this project has yielded at least a partial success.

#### V. References:

- Arrand, Helena Francis. Optical Waveguides and Components Based on Porous Silicon, University of Nottingham, September 1997. Chapter 3: "Optical Waveguiding Theory", Pp 18-35.
- Blok and Lenstra. "Modal Analysis of a Planar Waveguide with Gains and Losses." *IEEE Journal of Quantum Electronics* (Vol. 31, no 10) 1803-1809.
- Griffiths, David J. Introduction to Electrodynamics, Third Edition. (Prentice Hall, Saddle River, NJ, 1999). Ch. 9, "Electromagnetic Waves". Pp 364-411.
- Laughton, Marsh, and Button. "Time-Resolved Nonlinear Absorption Modulations in GaAs/AlGaAs Multiple-Quantum-Well Waveguides at 1.06 Microns." *IEEE Photonics Technology Letters* (Vol. 4, no 8) 863-864.
- Lee, Chung, Coldren, and Dagli. "On Leaky Mode Approximations for Modal Expansion in Multilayer Open Waveguides." *IEEE Journal of Quantum Electronics* (Vol. 31, no 10) 1790-1801.

Lee, Kost, et al. "Nonlinear Absorption Properties of AlGaAs/GaAs Multiple Quantum Wells Grown by Metalorganic Chemical Vapor Deposition."

Livescu, Miller, et al. "Free Carrier and Many-Body Effects in Absorption Spectra of Modulation-Doped Quantum Wells." *IEEE Journal of Quantum Electronics* (Vol. 24, no 8) 1677-1688.

Rao, Nannapaneni. Elements of Engineering Electromagnetics. (Prentice Hall) Ch. 7: "Waveguides". Pp 286-311.

Pedrotti and Pedrotti, Introduction to Optics, Second Ed. (Prentice Hall, Saddle River NJ, 1993).

Phillips and Harbor. Feedback Control Systems. (Prentice Hall, Saddle River, NJ, 2000).

```

%%%%%%%%%%%%%ALEXANDER ATANASIU%%%%%%%%%%%%%
%%%%%%%%%%%%%SWARTHMORE COLLEGE%%%%%%%%%%%%%
%%%%%%%%%%%%%ENGINEERING FINAL DESIGN PROJECT%%%%%%%%%%%%%
%%%%%%%%%%%%%MATLAB PLOTTING PROGRAM FOR%%%%%%%%%%%%%
%%%%%%%%%%%%%TRANSCENDENTAL EQUATION OF%%%%%%%%%%%%%
%%%%%%%%%%%%%LEAKY WAVEGUIDE%%%%%%%%%%%%%
%%%%%%%%%%%%%MAY 5, 2005%%%%%%%%%%%%%

```

```

%%%%%%%%%DEFINE CONSTANTS FOR THE WAVEGUIDE%%%%%%%%%

```

```

d=.5*10^-6;
h=.5*10^-6;
n0=1;
n1=1.3;
n2=1.4;
n3=1.5;

```

```

%%%%%%%%%SET LENGTH OF ARRAYS%%%%%%%%%

```

```

kx=0:20000:2000000;
transcendental=kx;
trnorm=kx;
treal=kx;
a0=kx;
a1=kx;
a2=kx;

```

```

%%%%%%%%%FIND VALUES FOR TRANSCENDENTAL FUNCTION FOR ALL VALUES OF KX%%%%%%%%%

```

```

for i=1:101
a0(i)=sqrt(-kx(i)^2+(n3^2-n0^2)*5.6*10^12);
a1(i)=sqrt(-kx(i)^2+(n3^2-n1^2)*5.6*10^12);
a2(i)=sqrt(-kx(i)^2+(n3^2-n2^2)*5.6*10^12);
transcendental(i)=(exp(-a2(i)*d)*(a1(i)-j*kx(i))/(j*kx(i)*(a1(i)+a2(i)))*((a2(i)+j*kx(i))+(a2(i)-
j*kx(i))*exp(2*j*h*kx(i))*exp(2*j*atan(kx(i)/a0(i)))));
trnorm(i)=norm(transcendental(i));
treal(i)=real(transcendental(i));
end

```

```

%%%%%%%%%PLOT TRANSCENDENTAL FUNCTION VERSUS KX, FIND A KX THAT FITS%%%%%%%%%

```

```

plot(kx, treal,'g',kx,2,'r',kx,trnorm,'b');
%plot(kx, transcendental,'g',kx,2,'r')

```

```

%%%%%%%%%%%%%ALEXANDERATANASIU%%%%%%%%%%%%%
%%%%%%%%%%%%%SWARTHMORE COLLEGE%%%%%%%%%%%%%
%%%%%%%%%%%%%ENGINEERING FINAL DESIGN PROJECT%%%%%%%%%%%%%
%%%%%%%%%%%%%MATLAB PLOTTING PROGRAM FOR%%%%%%%%%%%%%
%%%%%%%%%%%%%NORMAL MODES OF LEAKY WAVEGUIDES%%%%%%%%%%%%%
%%%%%%%%%%%%%MAY 5, 1995%%%%%%%%%%%%%

```

```

d=0.5*10^-6
h=0.5*10^-6
A=0
B=0
C=0
D=0
E=0
F=0
G=0
H=0
K=1

```

```

%%%%%%%%%Q SELECTED FROM ARRAY ELEMENTS OF KX THAT FIT TRANSCENDENTAL RELATION%%%%%%%%%
q=63

```

```

a21=a2(q)
a11=a1(q)
a01=a0(q)
kx1=kx(q)

```

```

%%%%%%%%%PLUG INTO INTERMEDIATE RESULTS TO SOLVE FOR COEFFICIENTS%%%%%%%%%

```

```

E=1
F=E*(a21-a11)/(a21+a11)
G=E*2*a21/(a21+a11)
C1=1
D1=(exp(j*2*atan(kx1/a01))*exp(j*2*kx1*h))
A1=exp(j*kx1*h)*(1+exp(j*2*atan(kx1/a01)))
K=(E*exp(-h*a21)+F*exp(h*a21))/(1+D1)
C=C1*K
D=D1*K
A=A1*K

```

```

%%%%%%%%%SELECT X-RANGES FOR EACH PIECEWISE FUNCTION%%%%%%%%%

```

```

x2=0:2*d/50:d;
x3=d:h/50:d+h;
x4=d+h:2*10^-8:d+h+10^-6;
x0=0:-2*10^-8:-10^-6;

```

```

%%%%%%%%%FIND Y VALUES FOR EACH X%%%%%%%%%

```

```

for i=1:51
y4=A*exp(-a01*(x4-d-h));
%y3=C*exp(-j*kx1*(x3-d))+D*exp(j*kx1*(x3-d));
y3=j*((C-D)*j)^2*cos(2*kx1*(x3-d))+(C+D)*2*sin(2*kx1*(x3-d))
%y3=0*x3
y2=E*exp(a21*x2)+F*exp(-a21*x2);
y0=G*exp(a11*x0);
end

```

```

%%%%%%%%%PLOT X, Y%%%%%%%%%

```

```

plot(x0,y0,'bp',x2,y2,'gp',x3,y3,'rp',x4,y4,'p')

```

## **Initial Mount Considerations from Existing Setup:**

### **Topic: Design setup for cooling the waveguide**

The goal of the design is to be able to take input/output voltage data for the waveguide when cooled to temperatures below freezing. Currently, data has only been taken for temperatures at and above room temperature, or approximately 23°C.

Some options were discussed, including liquid cooling using liquid nitrogen, forced fan cooling, and thermoelectric cooling. It was decided that we should proceed to investigate the thermoelectric cooler (TEC) method, as the parameters seemed to fit our requirements.

Various companies that carried TECs were found on the Internet, the Melcor Corporation was contacted as a design information base. A summary of the parameters are:

Ambient temperature: 23°C

Target temperature: -30°C

Active Load: 0 W

Size limitations: < 3"

Maximum voltage: 5V

Maximum current: 3A

Using these design parameters, an appropriate TEC was selected using Melcor's AZTEC software, as well as a suitable heat sink. It was discovered that a new power supply would have to be purchased as none of the current sources in the laboratory could supply the necessary power to the TEC.

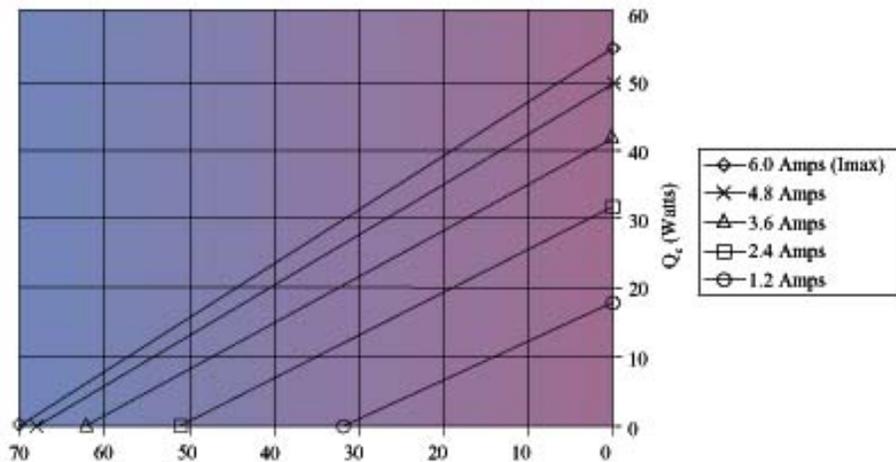
A design for the setup was then made, which involves the heat sink mounted on its side as the heat sink's efficiency would be affected by the mount. The aluminum mount for the waveguide had to be made as small as possible and insulated with foam to prevent heat gain from natural convection. A hole for the heater was also incorporated into the design however there are still concerns about the possibility of the heater melting the foam insulation around the mounting block.

The design was faxed to Melcor for final approval that it will operate as required.

## Thermoelectric Coolers: Overview and Summary

Thermoelectric Coolers (TEC's) use interfaces between dissimilar conductors to produce the "Peltier Effect", in which one side of the interface will reach a lower temperature than the other side. There is a net amount of heating between the hot and cold surface, because the power applied to the TEC exits as heat. However, the temperature gradient can be large enough to produce a large cooling effect on the cold side, especially if heat is efficiently removed from the hot side of the cooler. This temperature gradient is proportional to the power driven through the coolers, though this dependence becomes much smaller as this power nears its maximum value, leading to less efficient cooling which becomes especially pronounced with the rise in heat produced by this power.

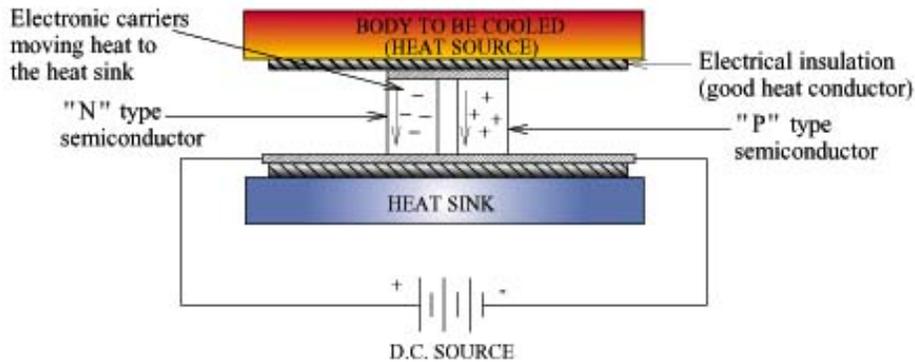
All thermoelectric coolers have a maximum theoretical temperature difference. They get less and less efficient as the temperature gradient gets higher, so that, even at the maximum power, they will reach a maximum temperature difference where they can no longer offer any cooling power.



**Fig. 1:** The Cooling Power (vertical axis) versus Temperature difference (horizontal) for a typical thermoelectric cooler. Since our cooler has two stages, the theoretical temperature difference would be significantly higher, and our unit would be able to deliver significant cooling power at over 50 Kelvin temperature difference, which was our initial goal. Note, that the cooling becomes much less effective higher in the power spectrum.

The maximum temperature difference for most thermoelectric coolers is below 60 K, and this maximum is never achieved, primarily because the cold side often needs a substantial amount of cooling power, which is only available at lower temperature differences. Moreover, the hot side of the cooler often gets substantially warm in relation to room temperature, so the actual temperature difference that can be achieved is less than it would be under ideal circumstances.

One method of resolving this problem involves the use of multi-stage TEC's formed by stacking coolers on top of one another. Even though each stage of the cooler can only reach a smaller temperature difference, these differences can add up, leading to substantially lower temperatures for the cold side than what could be achieved traditionally. The design of the waveguide mount involves two TEC's stacked on top of one another to produce a two-stage cooling.



**Fig. 2:** A circuit using a thermoelectric cooler would look like the one above. We see that the top cold plate in our system would be the waveguide mount. The electrical insulation used at the hot and cold interfaces was a thermal grease applied to the ceramic surfaces of the coolers. The TEC's were already equipped with unidirectional leads for the source voltage; this voltage could not be reversed to produce heating to the mount.

A circuit involving thermoelectric coolers is shown above. The performance of these circuits depends primarily on the efficiency of the heat sink at removing waste heat, the convective load of the cold side (heat source), and the quality of the thermal contact at the interfaces. Special care was made to improve these parameters in the waveguide mount design (see section III.G).

## LabView Documentation I:

**Innova 300 Power Scan**  
Software Documentation  
**Cameron Geddes**  
Swarthmore College  
1993

**Thomas R. Makin**  
Swarthmore College  
1996

### Introduction:

Innova 300 Power Scan is a program which controls a Coherent Innova 300 ion laser. It will scan the power of the laser up or down. It is pre-configured to collect data from 2 analog input channels on a National Instruments NB-MIO-16X input/output board in slot #4 of a Macintosh, and to write that data to a file. You could insert any data collection (or other) VI you want to in order to suit it to your needs.

The program was written in LabView 2.1.1, and requires that application to run. It also requires several sub-VIs, which are located in the folder "Innova 300 Power scan Sub-VIs", which is in the same folder as the program and this manual. Be sure to include all the sub-programs and this manual if you move or distribute this program. The program will warn you if there are any missing sub-VIs on startup.

### Setup:

The Innova 300 needs to be plugged into the printer port of your Mac. If you are networked with Appletalk, you will need to unplug the Appletalk cable from the printer port first, then go to the Chooser and select "inactive". Then restart your Macintosh and plug in the Innova into the printer port. If it isn't already, you need to configure your Innova to accept 300 baud communications (see the Innova 300 user manual). You can plug in up to 2 sensors from which data will be recorded at each pump increment. They should be plugged into the #4 and #5 channels on the National Instruments board. If you have changed the program by inserting another data collection VI, set it up as well.

### Operation:

Simply set the initial and final powers you want, and the increment (all in watts). The program will start at the value in "initial power" and scan up or down to the "final power." The "step pause" control sets the amount of time (in milliseconds) that the program waits after going to each new power level before taking data. You should allow enough time for the laser to stabilize (generally a few seconds). The "Number of Samples" control sets the number of readings taken from each sensor. If this is set greater than 1, the program will take that number of readings, average the readings, and write the average to file. This allows you to compensate for "noise" in your sensor, which is often detected by the very precise National Instruments boards even when no problem is shown with a standard multimeter. If you get bad data, you should try increasing the number of samples before anything else (100 is a good number). The "Filename" control sets the name of the file data will be logged to. The "Pathname" control sets the pathname of the folder into which the data file will be written. This needs to be a full pathname, including drive.

The data file will be a text file with columns of data separated by spaces. The first column of the file is the output of the sensor connected to channel 4, the second is the output of the sensor connected to channel 5., and the third column is ion laser power. You can open this file in most graphing programs, including Kaliedagraph and Cricket Graph.

**Distribution:**

Please distribute this program freely, but only when accompanied with all of its sub-VIs and documentation. No guarantees express or implied are given on whether this software works, and the user assumes responsibility for any risks or damages resulting from its use.

**Key:**

Make sure that data acquisition board is in the proper expansion slot. Currently, on a Quadra 650, the board is device #6. Make sure that connection to Innova 300 power drive is in port #1 (printer port). If either of these are incorrect, changes can be made either physically (moving hardware) or through changing the LabVIEW programs.

## LabView Documentation II:

**Temp Controller IV.**

Software documentation  
Cameron Geddes  
Swarthmore College  
1993

Thomas R. Makin  
Swarthmore College  
1996

Christina Erwin  
Bryn Mawr College  
1998

Recompiled and updated by:  
Alex Atanasiu  
Swarthmore College  
2005

### Introduction:

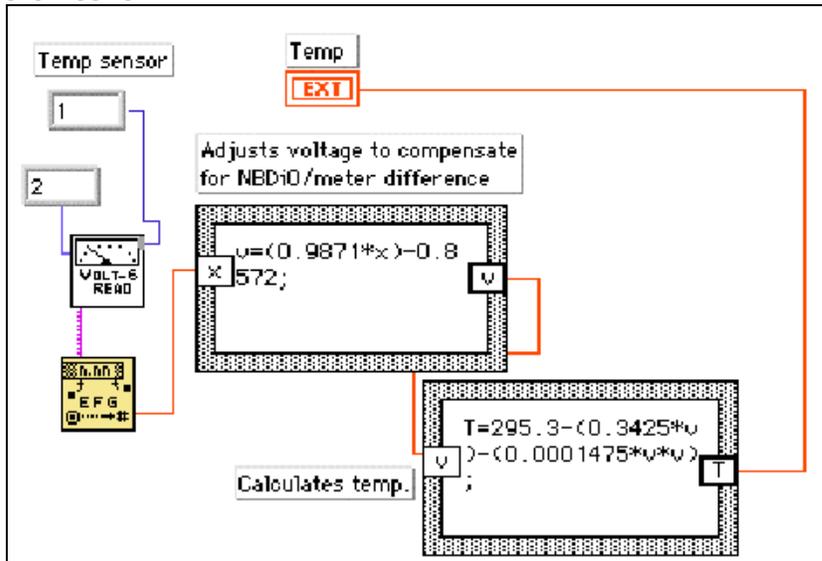
Temp Controller IV is a temperature control program written in LabVIEW 4.1. When installed, it can control the temperature of anything that has both a heater and a temperature sensor attached. It does so by controlling the heater's power supply through the Computer Controlled Power Supply. It cannot run alone, and requires the LabVIEW 4.1 application to run. It also requires the sub-VIs in the folder "TempControlVIs", which is located in the same folder as the program and these documents. Please be sure to take the Sub-VIs folder and the documents if you move Temp Controller anywhere. Temp Controller will notify you of any missing sub-VIs at startup time.

### Setup:

The program comes pre-configured for use with a Lakeshore 25 watt, one- $\Omega$  Cartridge Heater, and a Lakeshore DT-471-BR temperature sensing diode, capable of measuring accurately down to 10°K. The diode is driven by 0.395mA of current from a current supply. If you change sensors or the drive current, you will need to re-calibrate the *TempRead* VI so that it reads accurately with your setup. All this requires is an equation that converts the voltage read by the computer from your sensor to a temperature. This equation needs to be put into the *TempRead* VI (Fig. 1), and the unit will be ready to operate with your setup. Changing heaters will not normally necessitate a change in the program, but keep in mind that the Power Supply is rated only for 25W at 25V DC (see PS documentation), and that a less precise heater may ruin your accuracy.

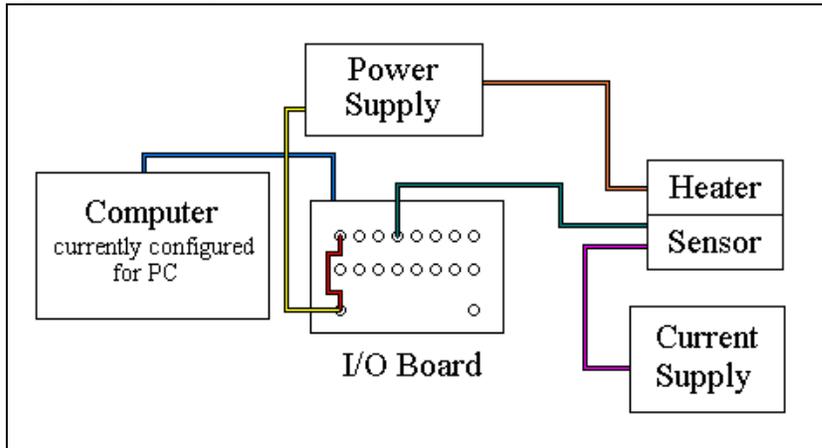
Fig. 1: The diagram of the *TempRead* VI.

Insert an equation in the upper left equation box to calibrate the I/O board with your multimeter, and an equation in the lower right box giving the relationship of temperature to voltage. You can get these equations by plotting data on voltage and temperature in KaleidaGraph or another graphing program.



Using the suggested configuration, with the Lakeshore units, your setup should look something like this:

Fig 2: Suggested Setup:



The computer is attached to the I/O board via their respective serial ports (blue wire). The In (BNC jack) terminal of the power supply is connected to the (DAC0OUT) Analog output of the I/O board (yellow wire). The "out" terminals of the Power supply are connected to the heater. The #0 Analog output is also connected to the #0 Analog input (red wire) so that the computer can monitor its own output. The Power supply is connected to the heater ("experiment") by the orange wire to the right, and the temperature sensor

is connected to the #3 Analog input by the green wire. The purple wire connects the current supply and the temperature sensor that it drives.

### How Temp Controller IV works:

Temp Controller IV works through a slope-evaluation method, which is a simplified version of a derivative controller. It takes two readings (separated by a period of time set by the “wait between samples” control) and evaluates the rate of change in the temperature. This is compared to the rate of change that would be desirable given the difference between the temperature being sought and the current temperature. The controller then adjusts the heater to increase or to decrease the rate as necessary. The equation that determines the desirable slope is set up so that a large slope is set for large  $\Delta T$ , and smaller slopes that result as the target is approached have a smaller  $\Delta T$ . The controller operates in three phases: seek, stabilize 1, and stabilize 2. The seek phase brings the temperature to within a predetermined range (set by the “Range” control) around the target (which is set with the “temp to seek” control). The “stabilize 1” function then takes over, and stabilizes the temperature over a time set by the user (using the “stabilization 1 time” control). Lastly, the “stabilize 2” function takes over, and further stabilizes temperature. If the “continuous stabilization” switch is on, this function will continue until the program is stopped or the switch is released. If it is not, the function will run for the time set by the “stabilization 2 time” control.  $K_1$ ,  $K_2$ ,  $K_3$ ,  $K_a$ ,  $K_b$ , and  $K_c$  are constants that define how the program’s seek equations function. The default values are:

$$\begin{array}{lll} K_1 = 20 & K_2 = 3 & K_3 = 0.5 \\ K_a = 40 & K_b = 30 & K_c = 3 \end{array}$$

You may want to change these values to optimize the program to the responsiveness of your heater/sensor.

**Pseudo-code equivalent:**

The Pseudo-code equivalent of the controller program is written out below, so that you can have a better idea of how it works:

```

Let desired temp =  $G1 = T_d$ 
Let  $G8 = \text{current } \Delta T / \Delta t$ 
Input:  $T_d$ 
      Range
       $K_1 \dots K_c$ 
      Stab. 1 time
      Stab. 2 time (or "continue stabilization")
      wait between samples
Let  $x = \text{wait between samples}$ 

WHILE:  $G7 < (G1 - \text{range})$  OR  $G7 > (G1 + \text{range})$ , do:
   $G6 \leftarrow \text{current temp}$ 
  wait  $x$  milliseconds
   $G7 \leftarrow \text{current temp}$ 
   $G8 \leftarrow (G7 - G6) / (x / 1000)$ 
  IF  $G8 < ((T_d - T) / K_1)$  then
     $G9 \leftarrow G9 + (\text{abs}(T_d - T) / K_1) * K_2 + K_3$ 
  IF  $G8 > ((T_d - T) / K_1)$  then
     $G9 \leftarrow G9 - (\text{abs}(T_d - T) / K_1) * K_2 - K_3$ 
  IF  $G9 \leq 25$  Volts then
    If  $G9 \geq 0.09$  then
      Write  $G9 \rightarrow \text{Power supply controller}$ 
    Else
       $G9 \leftarrow 0.09$ 
      Write  $G9 \rightarrow \text{Power supply controller}$ 
  ELSE
     $G9 \leftarrow 25$ 
    Write  $G9 \rightarrow \text{Power supply controller}$ 

```

Deleted:

ENDWHILE

```

WHILE  $t < \text{"stabilization 1 time"}$ , do:
   $G6 \leftarrow \text{current temp}$ 
  Wait  $x$  milliseconds
   $G7 \leftarrow \text{current temp}$ 
   $G8 \leftarrow (G7 - G6) / (x / 1000)$ 
  IF  $G8 < ((T_d - T) / K_1)$  then
     $G9 \leftarrow G9 + (\text{abs}(T_d - T) / K_1) * K_2 + K_3$ 
  IF  $G8 > ((T_d - T) / K_1)$  then
     $G9 \leftarrow G9 - (\text{abs}(T_d - T) / K_1) * K_2 - K_3$ 
  IF  $G9 < 25$  Volts then
    If  $G9 > 0$  then
      Write  $G9 \rightarrow \text{Power supply controller}$ 
    Else
      Write  $0 \rightarrow \text{Power supply controller}$ 

```

ENDWHILE

```

If "continuous stabilization" = no, then
  WHILE  $t < \text{"stabilization 2 time"}$ , do:
     $G6 \leftarrow \text{current temp}$ 
    wait  $x$  milliseconds
     $G7 \leftarrow \text{current temp}$ 

```

```

    G8 ← (G7-G6)/(x/1000)
    IF G8 < ((Td-T)/Ka) then
        G9 ← G9 + (abs(Td-T)/Ka)*Kb + (abs((Td-T)*Kc))
    IF G8 > ((Td-T)/Ki) then
        G9 ← G9 - (abs(Td-T)/Ka)*Kb - (abs((Td-T)*Kc))
    IF G9 < 25 Volts then
        If G9 > 0 then
            Write G9 → Power supply controller
        Else
            Write 0 → Power supply controller
    ENDWHILE

Else
    WHILE "continuous stabilization" = true, do:
        G6 ← current temp
        Wait x milliseconds
        G7 ← current temp
        G8 ← (G7-G6)/(x/1000)
        IF G8 < ((Td-T)/Ka) then
            G9 ← G9 + (abs(Td-T)/Ka)*Kb + (abs((Td-T)*Kc))
        IF G8 > ((Td-T)/Ki) then
            G9 ← G9 - (abs(Td-T)/Ka)*Kb - (abs((Td-T)*Kc))
        IF G9 < 25 Volts then
            If G9 > 0 then
                Write G9 → Power supply controller
            Else
                Write 0 → Power supply controller
        ENDWHILE

```

**Operation:**

To use the defaults, simply set the values you want for "temp to seek" and whether you want continuous stabilization or not, and run the program. Otherwise, modify the constants and values you wish changed, and modify the equations and sub-VIs as outlined above if you have installed a different sensor, then run Temp Controller. The "current temp..." indicators will allow you to monitor the progress and stability of the controller.

Temp Controller IV will operate in the background (you can hide LabVIEW, or just switch to another application) under Windows '95 on the PC (or [with slight variations] on the MAC with System 7.1). This will usually cause an approximately 0.1°C increase in the worst case error (deviation from desired temperature) of the controller. If you run programs that tie up the machine and prevent multi-tasking for significant time periods, the controller will not work at all, or will lose a lot of accuracy. Most word processing, graphing, and game software seems to be fine, though. Just test it out with the applications you want to run or check periodically to make sure temperature is being maintained.

**Specifications:**

Rated specs are with the Lakeshore 25W cartridge heater and Lakeshore DT-471-BR sensor diode. The test setup had the heater embedded in an aluminum block, around which was foam insulation on 5 sides, and one open side to allow cooling. The sensor was put between the aluminum block and the foam. You may get better accuracy if your heater and sensor are less separated.

Drift (T=35° to 100°C, Room temp = 22°C, 3 hours): ±0.25°C (worst case)  
: ±0.15° C (normal)

**Distribution:**

Please distribute this program freely, but only when accompanied with all of its sub-VIs and documentation. No guarantees expressed or implied are given on whether this software works, and the user assumes responsibility for any risks or damages resulting from its use.

**Key:**

Make sure that data acquisition board is in the proper expansion slot. Currently, on the PC, the board is device #1. Make sure that connection to the Innova 300 power drive is in port 0. If either of these are incorrect, changes can be made either physically (moving hardware) or through changing the LabVIEW programs.

## LabView Documentation II:

### Temp Controller PID II. Software documentation

Alexander Atanasiu  
Swarthmore College  
2005

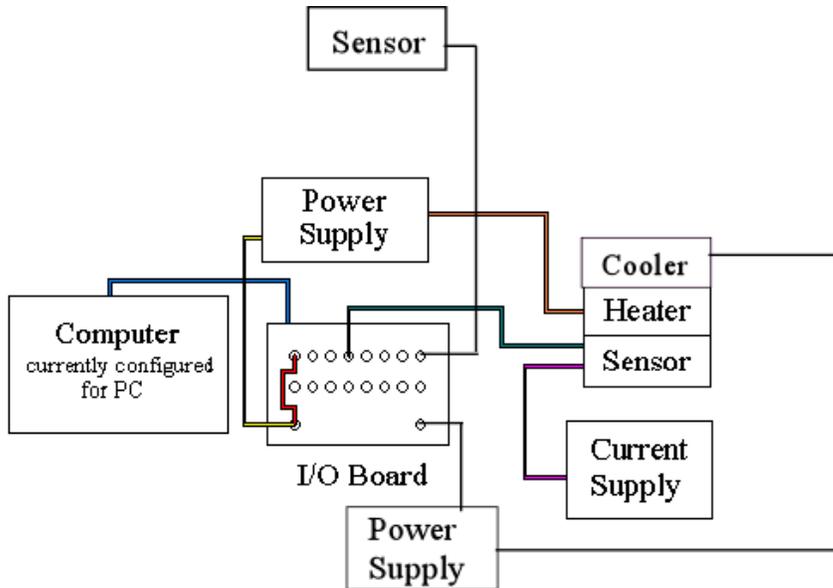
#### Introduction:

Temp Controller PID II is a temperature control program written in LabVIEW 6. This code automatically reads the room and mount temperatures from Lakeshore DT-471-BO sensors. It then controls the power to a resistance heater and a two-stage thermoelectric cooler using computer-controlled power supplies. It also requires the sub-VIs in the folder "TempControlVIs", which also contains the program itself. Temp Controller will notify you of any missing sub-VIs during the startup.

#### Setup:

The program comes pre-configured for use with a Lakeshore 25 watt, one- $\Omega$  Cartridge Heater, a two-stage unit built from Melcor CP 1.4-71-10L thermoelectric coolers driven in parallel, and two Lakeshore DT-471-BO temperature sensing diodes, capable of measuring accurately down to 10°K. The diode is driven by 10 microAmps of current from an op-amp circuit. If you change sensors or the drive current, you will need to re-calibrate the *TempRead* VIs so they read accurately with the current setup. Changing either power supply will necessitate a change in the *PS Controller* sub-VIs located in the *HeaterControllerPIDII* and *CoolerControllerPIDIII* VI's, in order to account for any gain or power limits that these new units might produce. It is not recommended to change either the heater or the cooler units themselves, as this would change the system response, and would require the user to redefine the PID gain coefficients.

The recommended setup is similar to that of the Temperature Controller IV vi (above), with a second power supply connected to the DAC 1 port to power the Coolers, and a second temperature sensor, which is read from Channel 7 on the DAQ board. The mount temperature is still read from Channel 3, and the heater power supply is still connected to DAC 0.



**Fig. 1: Recommended Setup:** This setup is the same as in Temp. Controller IV with the addition of the thermoelectric cooler, a second computer-controlled power supply, and a second temperature sensor to control room temperature.

### How Temp Controller PID II works:

The user will input a desired set temperature, which will be read by the program. The Temp Controller will automatically choose to either cool or heat the mount based on whether the desired temperature is above or below room temperature, as read by the second sensor, and use either the *HeaterController* or the *CoolerController* sub-VI to set to and maintain this temperature.

These sub-VIs each implement a standard PID controller with the default values:

#### HEATER

$$K_p = 0.14 \quad K_i = 0.02 \quad K_d = 0.15$$

#### COOLER

$$K_p = 0.3 \quad K_i = 0.03 \quad K_d = 0.33$$

These coefficients are multiplied by the (discrete time) values of the proportional, derivative, and integral error signals to produce a slightly overdamped response. This response is system-specific; if either the heater or cooler are replaced, or the mount is significantly altered, new PID constants will need to be found to replace the existing ones.

These sub-VI's also output a trigger signal to the main program that signals when the desired temperature is achieved. This trigger response will be set to low (0) if the temperature is far away from the set temp., but it will be high (1) when the temperature is within .2 degrees of the set temperature (this is the precision

limit of the temperature sensors themselves in our setup). The trigger is also limited by the derivative of the temperature signal, which would limit the possibility of a false positive in the event of a misreading.

**Operation:**

To use the default system, the user must set the desired values for “set temp” and set the values of *Glbl8.vi* and *Glbl1.vi* to zero before running the program. This latter task is accomplished by opening the global variables in Labview, setting the mode to “write”, setting x to zero, and running the programs once.

If the program is moved to a considerably faster or slower operating system, then the program may not run in the same amount of time, and the “delta T” value may need to be changed (default is 2 seconds). The “current temp” indicators will allow you to monitor the progress and stability of the controller.

Temp Controller IV will operate in the background, and will work with other programs running. If you run programs that tie up the machine and prevent multi-tasking for significant time periods, though, the controller will not work at all, or will lose accuracy. A fair amount of ringing may result, and the trigger may go off before steady state is achieved.

**Specifications:**

Rated specs are with the Lakeshore 25W cartridge heater, two Melcor CP1.4-71-10L TEC modules, a 12V, manually operated CPU fan and Lakeshore DT-471-BO sensor diodes. The TECs were insulated to prevent condensation, but the mount itself was air-cooled and uninsulated. Insulating the mount with foam, and water-cooling the hot side of the TEC modules may produce better results.

Drift (T=0<sup>o</sup> to 100<sup>o</sup>C, Room temp = 21.5<sup>o</sup>C, 1 hour): ±0.2<sup>o</sup>C (worst case)  
: ±0.1<sup>o</sup> C (normal)

**Distribution:**

Please distribute this program freely, but only when accompanied with all of its sub-VIs and documentation. No guarantees expressed or implied are given on whether this software will work, and the user assumes responsibility for any risks or damages resulting from its use.

**Key:**

Make sure that data acquisition board is in the proper expansion slot, with each power supply and temperature sensor connected to the proper port. Currently, on the PC, the board is device #1. If the setup is incorrect, changes can be made either physically (by moving hardware) or by changing the channels listed in the LabVIEW programs.