Experimental Design for Estimating Electro-Thermophysical Properties of a Thermopile Thermal Radiation Detector

by

Joel Barreto

Thesis Submitted to
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements
for the degree of

Master of Science

in

Mechanical Engineering

Dr. J. Robert Mahan, Chair
Dr. Thomas E. Diller
Dr. Brian Vick

July 16, 1998

Blacksburg, Virginia

Keywords: Experimental design, Thermal radiation detector,
Composite medium, Genetic algorithm

Experimental Design for Estimating Electro-Thermophysical Properties of a Thermopile Thermal Radiation Detector

by

Joel Barreto

Committee Chairman: Dr. J. Robert Mahan
Mechanical Engineering

(ABSTRACT)

As the Earth's atmosphere evolves due to human activity, today's modern industrial society relies significantly on the scientific community to foresee possible atmospheric complications such as the celebrated greenhouse effect. Scientists, in turn, rely on accurate measurements of the Earth Radiation Budget (ERB) in order to quantify changes in the atmosphere. The Thermal Radiation Group (TRG), a laboratory in the Department of Mechanical Engineering at Virginia Polytechnic Institute and State University, has been at the edge of technology designing and modeling ERB instruments.

TRG is currently developing a new generation of thermoelectric detectors for ERB applications. These detectors consist of an array of thermocouple junction pairs that are based on a new thermopile technology using materials whose electro-thermophysical properties are not completely characterized.

The objective of this investigation is to design experiments aimed at determining the electro-thermophysical properties of the detector materials. These properties are the thermal conductivity and diffusivity of the materials and the Seebeck coefficient of the thermocouple junctions. Knowledge of these properties will provide fundamental information needed for the development of optimally designed detectors that rigorously meet required design specifications.

Acknowledgments

I would like to express my deepest gratitude to all the people who in one way or another contributed to the completion of this work.

- To Dr. Robert Mahan, my thesis advisor, for motivating me to enroll at Virginia Tech, for his guidance and recommendations.
- To Dr. Thomas E. Diller and Dr. Brian Vick for accepting to serve on my advisory committee and for their suggestions and time devoted to my thesis.
- To the members of the Thermal Radiation Group, my lab partners, for their friendship and support during my graduate studies.
- To my wife for her continuous encouragement, help and patience.
- To my family for their support and for sharing with me the satisfaction of reaching this goal.

To all of you, Thank You!

Table of Content

List of Tables

List of Figures

Nomenclature

A_{in}, B_{in}	Coefficients of Ψ_{in}'s general solution
C_n	Coefficient of $T_{t,i}$'s general solution
c	Specific heat (J/kg$\cdot$K)
E_{ab}	Seebeck electromotive force for combination of thermoelements a and b (V)
E_i	Estimated property value
$\overline{E}$	Sample mean of the estimated property values
I	Electrical current (A)
j	Current density (A/m^2)
k	Thermal conductivity (W/m$\cdot$K)
M	Total number of layers
m	Number of observations
n	Index number of β, Number of Samples
n^+	Normalized density function
$n_{\max}$	Maximum value of density function
l	Layer thickness (m)
P	Power absorbed or liberated (W)
$P^{'}$	Power absorbed or liberated per unit length (W/m)
q''	Heat flux (W/m^2)
Q	Net heat flow per unit volume per unit time (W/m^3)
R	Electrical resistance (V/A)

S	Sum-of-square function
T	Absolute temperature (K)
T_a	Temperature of active junction (K)
T_{calc}	Calculated temperature (K)
T_i	Transient temperature (K)
T_{meas}	Measured temperature (K)
T_r	Temperature of reference junction (K)
$T_{s,i}$	Steady-state part of T_i (K)
$T_{t,i}$	Transient part of T_i (K)
ΔT	Junction temperature difference (K)
t	Time (s)
x, y	Cartesian coordinates (m)
x_i	Layer interface (m)
$\mathbf{0}$	Zero vector

Greek

α	Thermal diffusivity of layer i (m^2/s)
α_{ab}	Seebeck coefficient for thermoelements a and b (V/K)
α_w	Optimum wedge angle
β_n	Eigenvalue solution
Γ	Time-dependent function

δ_{in}	Algebraic simplification
η_E	Experimental observation
η_T	Predicted value
μ	Micro
ξ	Square matrix
π	Peltier coefficient (V/K)
ρ	Mass density (kg/m^3)
ρ_e	Electrical resistivity $(\mathrm{ohm} \cdot \mathrm{m})$
σ	Thomson coefficient (V/K)
τ	Time constant (s)
Ψ	Space-dependent function

1. Introduction

The industrial progress of humankind has influenced the global climate, including the nature of the Earth's surface and the composition of the atmosphere. For many years, the accumulation of greenhouse gases has significantly changed the chemical composition of the atmosphere. Greenhouse gases such as CO_2 and H_2O are produced by industry, particularly by the burning of fossil fuels. As the amount of these gases increases in the atmosphere, the amount of longwave radiation emitted by the Earth's surface and absorbed by the atmosphere is also increased. As a consequence, many scientists speculate that increasing the concentration of greenhouse gases could lead to significant global warming. This has influenced the scientific community to recognize the importance of quantifying the radiation balance within the Earth-atmosphere system.

Scientists rely on accurate Earth radiation budget (ERB) measurements in order to understand and quantify climate phenomena such as the greenhouse effect. The ERB is normally expressed in terms of the global energy balance at the top of the Earth's atmosphere. The most relevant ERB components are the incoming solar radiation, the reflected solar radiation, and the thermal radiation emitted by the earth and its atmosphere.

Measurements of the ERB components can only be obtained by using reliable radiation detectors in instruments such as radiometers on board an orbiting platform.

A new thermal radiation detector concept for Earth radiation budget radiometry applications is currently under development at the Virginia Polytechnic Institute and State University (Virginia Tech). This detector is based on a new thin-film thermopile technology. The objective of this investigation is to design experiments aimed at determining the electro-thermophysical properties of the detector materials under the specified environmental conditions. In this chapter, the reader is introduced to the new concept of thermal radiation detectors by briefly summarizing the present state of knowledge pertaining to the application, fabrication and design of the detector.

1.1 Radiometry

The detector considered in this investigation was originally intended to be used in a radiometer on board a geo-synchronous satellite whose rotation allows the detector to scan the Earth, as shown in Figure 1.1 [Muller, 1997; Oliver, 1998]. Specifically, a radiometer is an optical system, perhaps including one or more filters, and a thermal radiation detector used to measure the intensity of incident radiation within a specified wavelength interval and within a specified field of view. The components of a generic radiometer are illustrated Figure 1.2. It uses an optical system to gather the radiation from the Earth scene and to focus it upon the aperture of a detector cavity. The detector produces an electrical signal proportional to the radiance. A characteristic feature of an ERB radiometer is that the radiance received from the Earth scene is compared to one or more reference sources, e.g. space or internal sources, for calibration purposes.

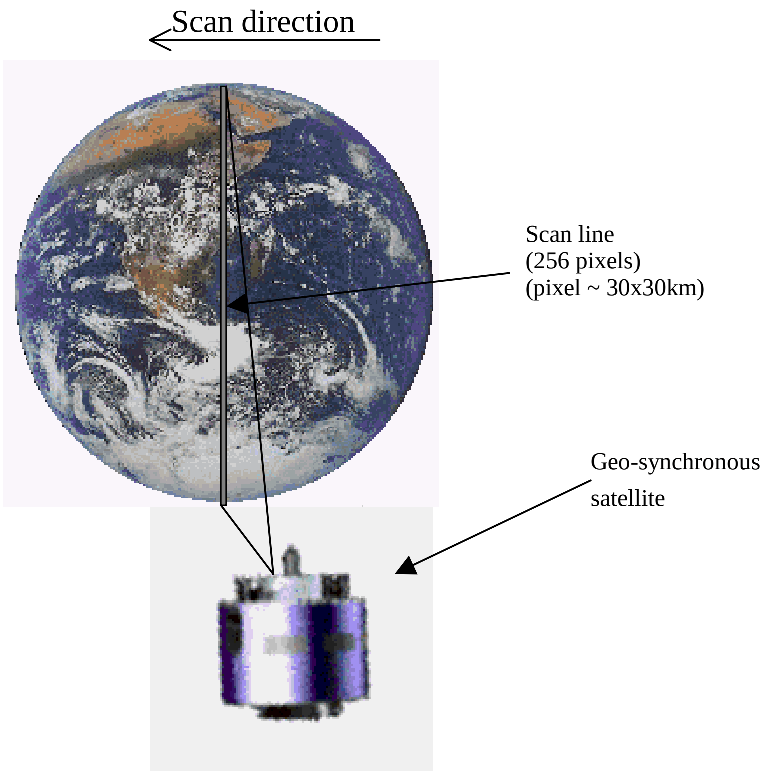


Figure 1.1 Detector scanning the Earth from a geo-synchronous satellite [Adapted from Muller, 1997].

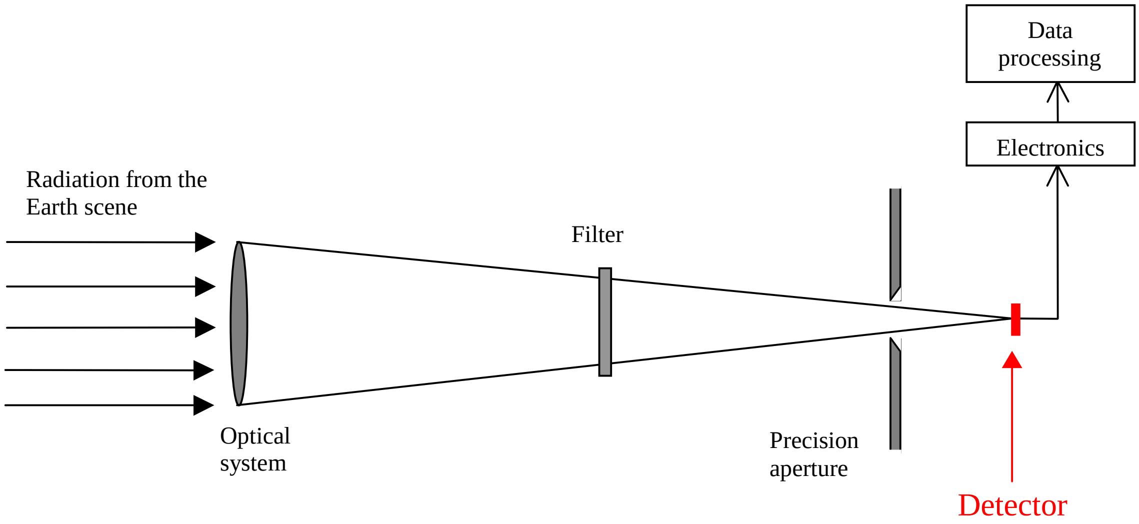


Figure 1.2 Schematic of a generic radiometer.

1.2 Thermopile Thermal Radiation Detector

Thermal radiation detectors are devices fabricated from substances that are sensitive to thermal radiation over a specified wavelength interval and that generate an electrical signal with strength related to the band-limited input radiance. The new concept of thermal radiation detectors includes thin-film technology and takes advantage of a unique combination of materials to enhance performance. The following sections describe the fabrication of the detector and the functional description of the detector design.

1.2.1 Design of the Sputtered Thermal Radiation Detector

The proposed GERB thermal radiation detector consists of a linear array of 256 thermocouple junctions. Each of these thermocouples, or thermoelectric circuits, is composed by a pair of junctions and is designed to produce a small output voltage proportional to the temperature difference between the junctions. A single thermocouple of the thermal detector, shown in Figure 1.3, consists of an absorber layer (Chemglaze), an active junction (sputtered zinc-antinomide and platinum layers), a thermal impedance (vapor-deposited parylene), and a reference junction (sputtered zinc-antinomide and platinum layers), assembled on an aluminum nitride substrate. The primary function of the absorber layer at the top of the active junction is to absorb the incident thermal radiation, while the primary function of the thermal impedance layer at the bottom is to establish the sensitivity of the detector. The junctions are formed between the layers of platinum and zinc-antinomide, a combination that increases by a factor of several hundred the Seebeck coefficient over traditional metal-metal junctions [Krieder, 1994]. In order to assure a high degree of reliability and invariant results compared to the thermocouple calibrations, one of the junctions, the reference junction, is maintained at a fixed temperature while the other junction, the active junction, assumes a temperature whose value depends on the amount of illumination. During normal operation of the detector, the absorber is directly exposed to the radiance arriving from the Earth scene, while the reference junction is entirely shielded and maintained at a fixed temperature. As heat is absorbed in the absorber layer, a temperature difference is developed across the thermal impedance layer, producing a temperature increase of the active junction. This temperature increase creates a temperature difference between the active and reference junctions that generates an electromotive force (emf) related to the magnitude of input radiance. The magnitude of the emf is directly proportional to the heat absorbed by the active junction, but it is also sensitive to the selection of the absorber, junction and thermal impedance material properties.

1.2.2 Fabrication of the Sputtered Thermal Radiation Detector

A thin-film polymer layer is vapor deposited on one half of one side of an aluminum nitride substrate base, as shown in Figure 1.4 (a). Next, two parallel bands of platinum are sputter deposited, one directly on the aluminum nitride substrate and the other on the thin film polymer layer, as shown in Figure 1.4 (b). Then a zinc-antinomide layer is applied to bridge the gap between the two platinum bands, as shown in Figure 1.4 (c). The platinum/zinc-antinomide junction on the thin-film polymer layer is then coated with a black absorber layer. Finally, a laser slices through the layers in order to form the linear-array of thermocouples, as shown in Figure 1.4 (d). The nominal properties of the materials used in the detector are given in Table 1.1. The purpose of this thesis is to define an *in situ* protocol for determining the actual values of these electro-thermophysical properties.

1.3 Integration Concept of the Thermal Radiation Detector

The sensor integration concept for the proposed GERB instrument consists of a linear array of thin-film thermocouples mounted on the wall of a wedge-shaped cavity, as shown in Figure 1.5. The walls of the cavity are made mirror-like over the wavelength interval of interest (0.1 to 40 μm). That is, the walls are intended to have a low absorptivity and a high specular reflectivity. On the other hand, the absorber layer on the thermopile active junction has a high absorptivity and a low reflectivity, but the reflectivity is mostly specular. As the collimated beam of radiation enters the aperture, most of the radiation is immediately absorbed by the linear array of thin-film thermal radiation detectors. The purpose of the cavity is to return the radiation reflected from the detector to the same element, or pixel, of the array. Hence, the multiple reflections between the absorber layer and the mirror-like walls create a "cavity effect" that significantly increases the effective absorptivity of the detector. Sánchez [1998] used the Monte-Carlo ray-trace (MCRT) method to determine the optimum wedge angle α_w at which the effective absorptivity is maximized and the optical crosstalk between the elements of the detector array is minimized.

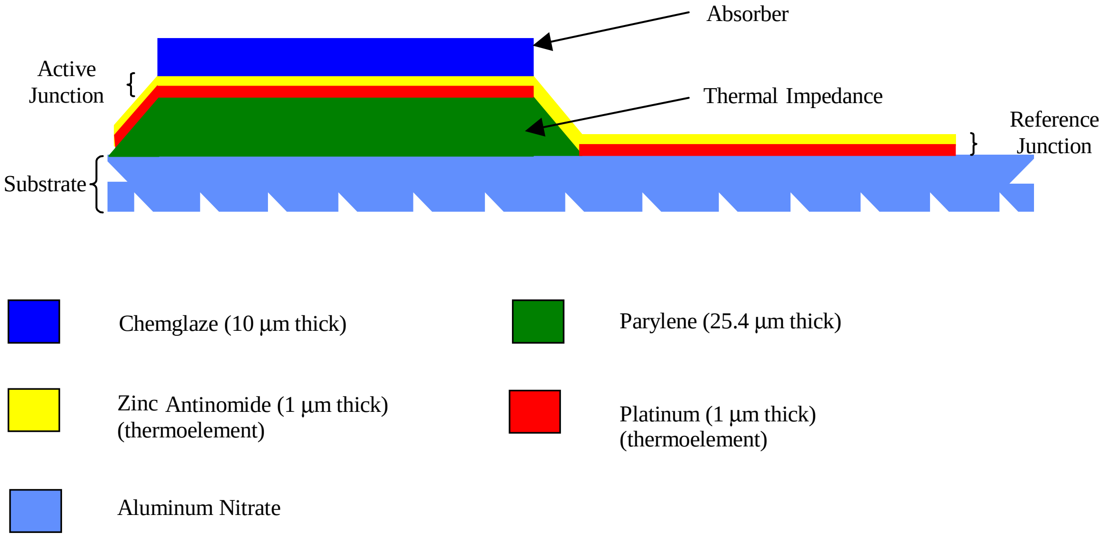
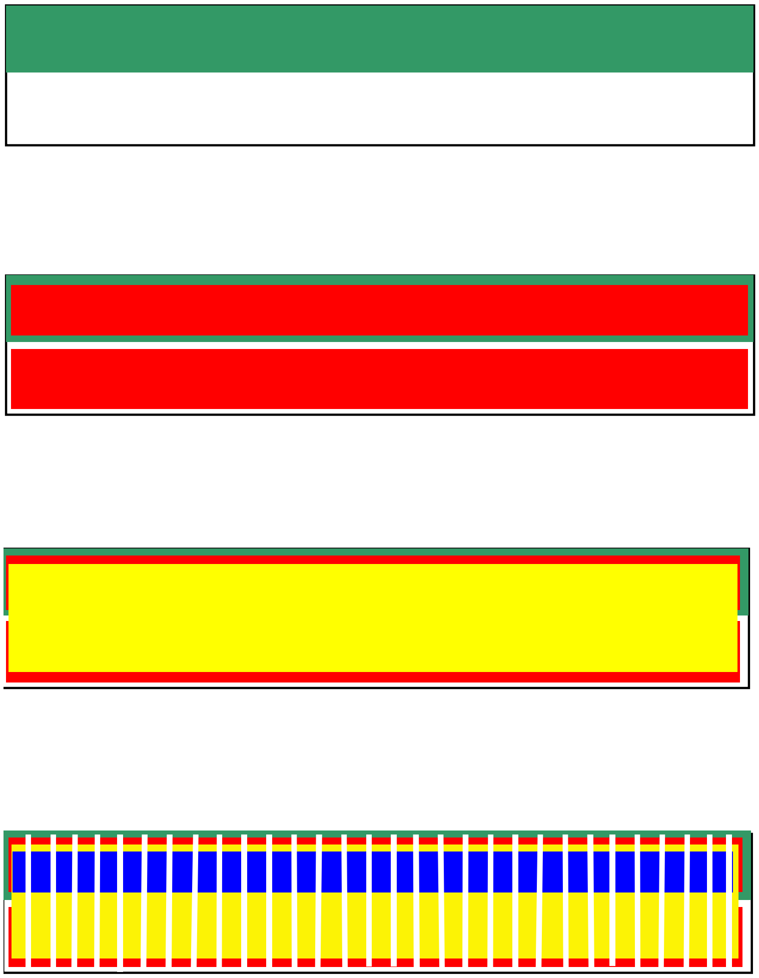


Figure 1.3 Cross-section of a single element of the thermal radiation detector (not to scale).

Table 1.1 Nominal properties of materials used in thermal radiation detector [Weckmann, 1997].

	Mass density (kg/m^3)	**Specific heat** (J/kg$\cdot$K)	**Conductivity** (W/m$\cdot$K)	**Diffusivity** (m^2/s)	**Seebeck coefficient** (V/K)
Aluminum nitride	3260	800	165	6.33×10^{-5}	N/A
Chemglaze	1400	669	0.209	2.23×10^{-7}	N/A
Parylene	1289	712	0.084	9.15×10^{-8}	N/A
Platinum	21450	133	71.6	2.51×10^{-5}	N/A
Zinc Antimonide	6880	200	60	4.4×10^{-5}	N/A
Platinum/Zinc -Antinomide junction	N/A	N/A	65.3	3.09×10^{-5}	920×10^{-6}



(a) Parylene thermal insulation (green) deposited onto one half of aluminum nitride substrate (white).

(b) Platinum (red) bands are sputtered on parylene band and aluminum nitride substrate.

(c) Zinc-antinomide (yellow) is sputtered to form junctions with the two platinum bands.

(d) Active junction coated with an absorber layer (blue) and laser sliced into pixels.

Figure 1.4 Fabrication steps of the sputtered thermal radiation detector [Mahan and Langley, 1996].

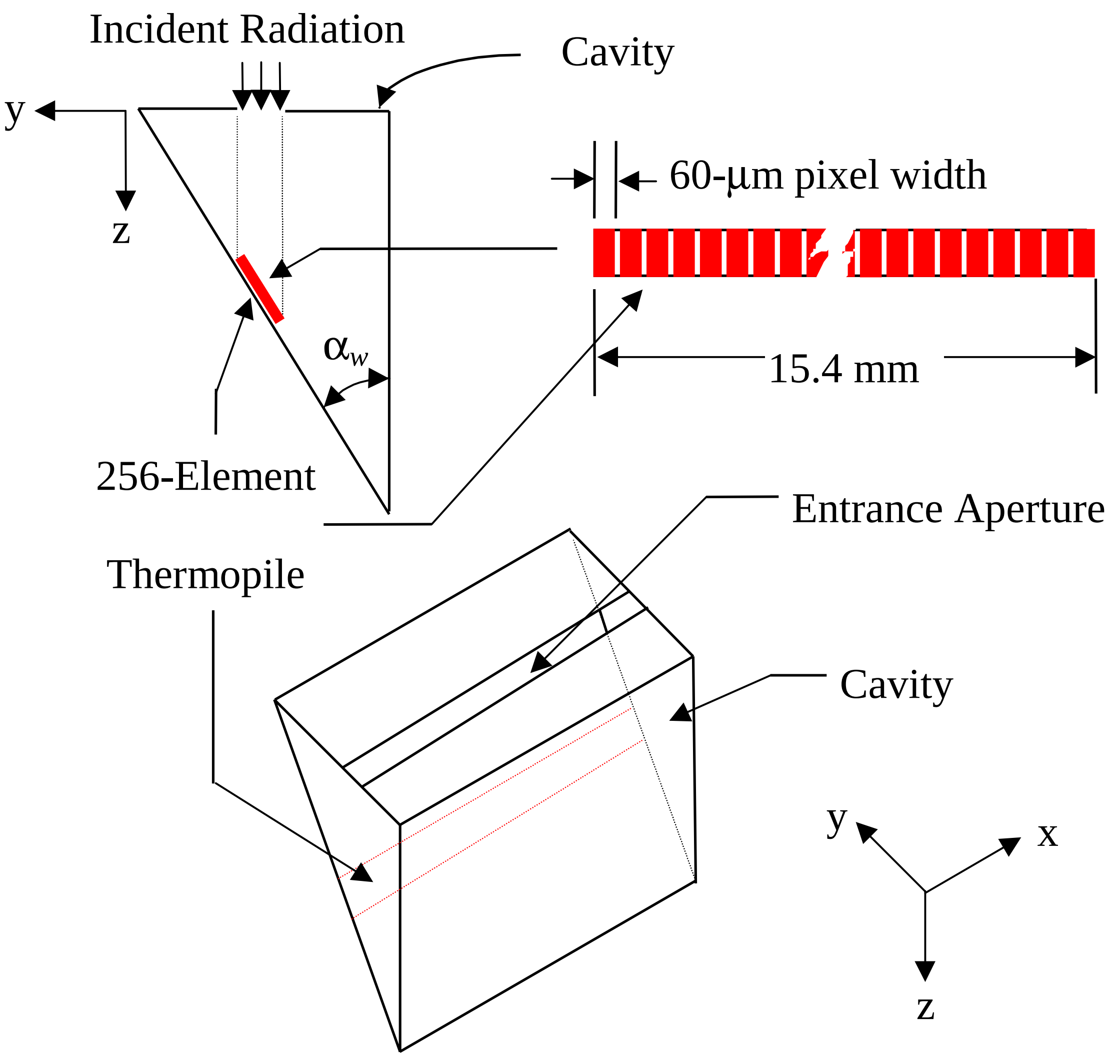


Figure 1.5 Integration concept of thin-film thermopile linear-array radiation detector and wedge-shaped cavity [Sánchez, 1998].

1.4 Motivation and Goal

A new thermal radiation detector concept for Earth Radiation Budget has been proposed by Dr. J. R. Mahan, professor in the Department of Mechanical Engineering at Virginia Tech, and Mr. L. W. Langley, president of Vatell Corporation. A combined effort between the Thermal Radiation Group in the Mechanical Engineering Department at Virginia Tech, Vatell Corporation, and NASA's Langley Research Center has been initiated in the design and development of the detector.

The Thermal Radiation Group, under the supervision of Dr. Mahan, has been actively contributing to the detector design and modeling in the last two years. Recently, Weckmann [1997] created a model of the dynamic electrothermal response of the thermal radiation detector. Sánchez [1998] created a Monte-Carlo ray-trace radiative model of the integrated detector concept intended to be used in the European Geostationary Earth Radiation Budget (GERB) instrument.

The present investigation is aimed at designing experiments capable of characterizing the electro-thermophysical properties of the candidate detector materials. Because of the thin-film manufacturing technology used to fabricate the thermal detector, electro-thermophysical properties may differ significantly from bulk properties [Lambropoulos, *et al*., 1991]. Accurate estimation of the electro-thermophysical properties of the detector materials is extremely important in the design and development of the actual detector. Therefore the material properties need to be measured in the context (thickness and manufacturing process) of the actual detector.

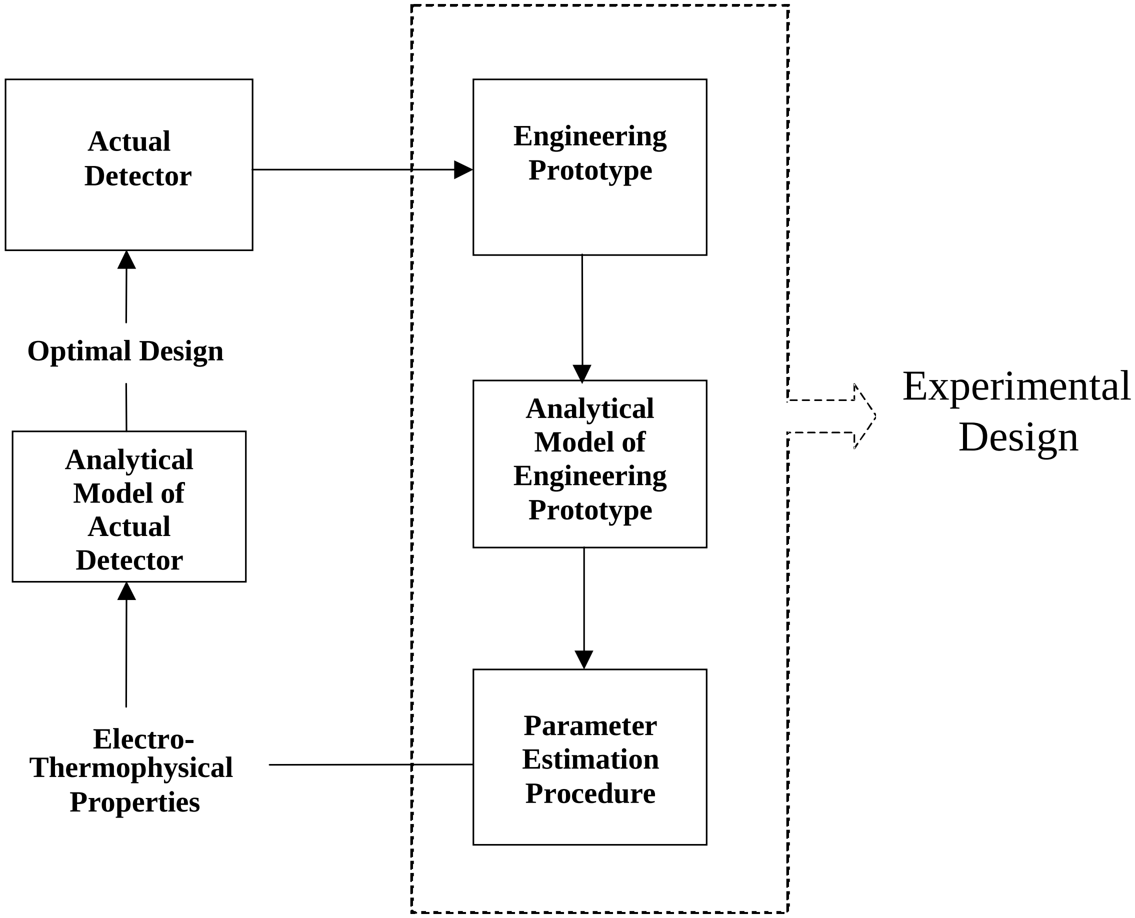

Actual Detector
Engineering Prototype
Optimal Design
Analytical Model of Engineering Prototype
Experimental Design
Analytical Model of Actual Detector
Electro-Thermophysical Properties
Parameter Estimation Procedure

Figure 1.6 Logic flow of actual detector design and development.

The key entities of the design and development of the actual detector are shown in Figure 1.6. Knowledge of the electro-thermophysical properties of the actual detector will provide fundamental information needed for the development of optimally designed detectors that rigorously meet required design specifications. The estimation of these properties can be accomplished by the development of an "experimental design". The components of the experimental design describe the objectives of this investigation. These objectives are:

1. to develop an engineering prototype detector that will permit efficient and accurate property estimation of the materials used the actual detector,

2. to develop a mathematical model of the engineering prototype that will efficiently predict the heat conduction within the experimental set-up, and
3. to determine an parameter estimation procedure that estimates the electro-thermophysical properties of the detector materials based on experiments involving the experimental set-up.

The thermal properties considered in this investigation are the conductivity and diffusivity of each of the materials of the detector. The thermoelectric property considered is the absolute Seebeck coefficient, which strictly depends on the combination of the thermoelectric materials used in the detector.

2. Literature Review

This chapter summarizes the accumulated knowledge related to the estimation of electro-thermophysical properties. The first section introduces the principles governing the operation of thermocouple devices. The reader is introduced to the phenomena involving an interconversion of heat and electrical energy; specifically, the origin and nature of the Seebeck, Peltier and Thomson thermoelectric effects. The relationship of these thermoelectric effects is consequently described by the thermodynamics of thermoelectricity effects.

The second section presents the analytical heat conduction analysis of a one-dimensional composite medium including both steady-state and transient heat conduction processes. The purpose of this analysis is to develop a mathematical model that will be used to predict the heat conduction behavior of the engineering prototype of the actual thermal detector.

2.1 Thermocouple Description

For more than a century thermocouples have been used extensively for temperature measurement. A thermocouple is a thermoelectric circuit consisting of two dissimilar metals, or thermoelements [Pollock, 1982]. The corresponding interfaces between the two thermoelements are called junctions.

Thermocouples normally produce a voltage output that varies in a predictable way with the temperature difference between junctions. If a temperature difference is applied between the junctions of a thermocouple, then either an electromotive force can be detected by a voltmeter, or an ammeter can detect a current. In order to produce invariant results compared to the thermocouple original calibrations and therefore a high degree of reliability, one of the junctions, the reference junction, is maintained at a fixed temperature while the other junction, the measuring junction, is held at the unknown temperature to be determined. By using this combination of a reference junction and a measuring junction, the magnitude of electromotive force becomes a function of the temperature difference, which is directly dependent on the measuring junction temperature and thermoelement materials.

Three thermoelectric effects have been identified: the Seebeck effect, the Peltier effect and the Thomson effect. The following three sections describe in some detail the characteristics of each of the thermoelectric effects. The thermodynamics of thermoelectricity is explained in a subsequent section.

2.1.1 The Seebeck Effect

In 1821 Thomas J. Seebeck discovered that a small electric current could be produced in a closed circuit composed of two different conductor materials (a and b) if the junctions are maintained at different temperatures [Tauc, 1961]. This is illustrated in Figure 2.1.

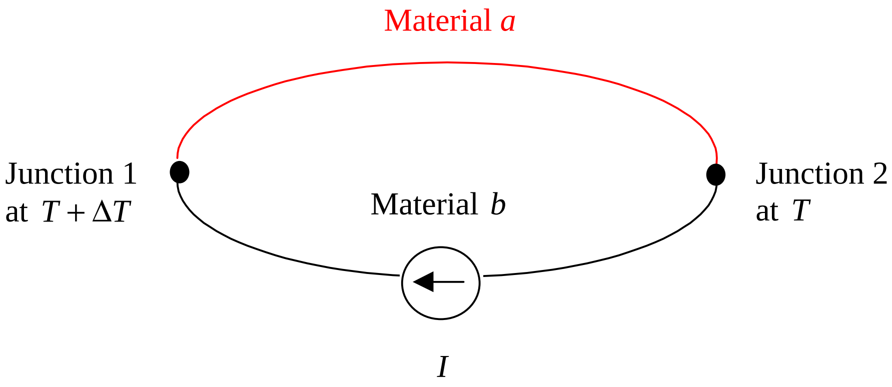


Figure 2.1 Seebeck effect in a closed circuit.

In the case of an open-circuit, an electromotive force, or emf E_{ab} (V), is produced across the terminals of a break, as shown in Figure 2.2. The open-circuit voltage, E_{ab} (V), is proportional to the junction temperature difference ΔT (K) and is described as

$$E_{ab} = \alpha_{ab}(T)\,\Delta T\,, \qquad (2.1)$$

where α_{ab} (V/K) is the Seebeck coefficient. The Seebeck coefficient depends upon the selection of the conductor materials, and is defined

$$\alpha_{ab}(T) = \lim_{\Delta T \to 0} \frac{\Delta E_{ab}}{\Delta T} = \frac{dE_{ab}}{dT}\,, \qquad (2.2)$$

where $\frac{dE_{ab}}{dT}$ is commonly referred to as the thermoelectric power, the change in emf per degree Kelvin. An absolute Seebeck coefficient is normally assigned to conductor materials in such a way that the Seebeck coefficient is defined as the difference between the absolute Seebeck coefficients of the conductor materials combined to form a thermocouple.

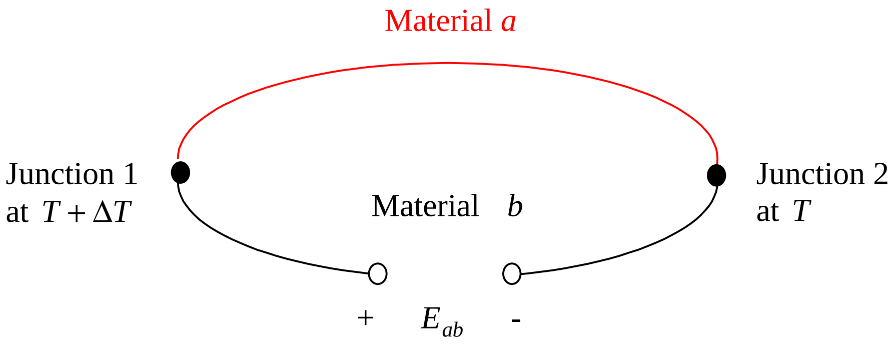


Figure 2.2 Seebeck effect in an open circuit.

2.1.2 The Peltier Effect

In 1834, Jean Charles Athanase Peltier discovered that when an electric current flows through the junction of two dissimilar materials, as shown in Figure 2.3, heat was either liberated or absorbed at the junction [Tauc, 1961]. The rate of absorption or liberation of heat P (W) at the junction is directly proportional to the electric current I (A) passing through the conductor materials, and is quantitatively described as

$$P = \pi(T)\, I\,, \tag{2.3}$$

where $\pi(T)$ (V) is the proportionality coefficient commonly known as the Peltier coefficient.

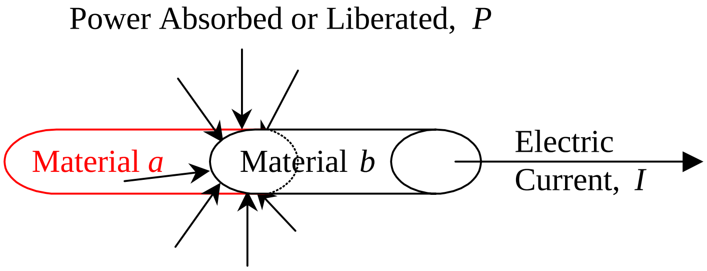


Figure 2.3 Illustration of the Peltier effect.

The direction of current flow determines whether the heat is absorbed or liberated. In the case of electric current flowing in the same direction as the Seebeck current, heat is absorbed at the hotter junction and liberated at the colder junction. This reversible heating and cooling process is independent of the shape or dimensions of the conductors at the junction. That is, the Peltier effect is not associated with the nature of the contact between the two conductors [Pollock, 1982].

2.1.3 The Thomson Effect

In 1854 William Thomson discovered that an energy interchange occurs with the surroundings, as illustrated in Figure 2.4, when an electric current flows through a conductor in the presence of a temperature gradient [Tauc, 1961]. In other words, the Thomson effect influences the temperature of the conductors between the junctions rather than the junctions themselves. The power absorbed or liberated per unit length is proportional to the electric current and the temperature gradient, and is quantitatively described by

$$P' = \tau(T)\, I\, \frac{dT}{dx}, \qquad (2.4)$$

where $\tau(T)$ (V/K) is the coefficient of proportionality that is commonly known as the Thomson coefficient.

The direction of current flow determines whether the heat is absorbed or liberated. In the case of electric current flowing in the direction of the negative of the temperature gradient, heat is liberated along the conductor, but heat is absorbed if the electric current flows counter to the negative of temperature gradient.

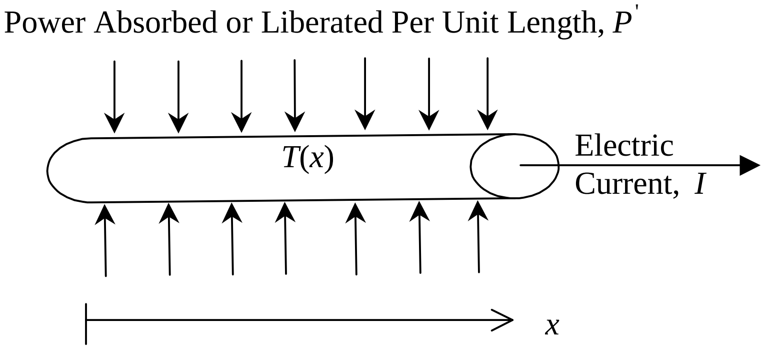


Figure 2.4 Illustration of the Thomson effect.

The reversible Thomson effect should not be confused with the irreversible Joule effect which describes the heat generation, I^2R, produced by a current flowing through a conductor without temperature gradient along its length. When considering both the Joule and Thomson effects, the net heat liberated or absorbed per unit volume per unit time, Q (W/m·s), in a conductor is

$$Q = j^2 \rho_e - \tau(T)\ j\,\frac{dT}{dx}, \qquad (2.5)$$

where j (A/m^2) is the current density and ρ_e (Ω·m) is the electrical resistivity of the conductor [Pollock, 1982].

2.1.4 Thermodynamics of Thermoelectricity Effects

Since the nineteenth century, the fundamental principles of thermoelectricity have been understood on a macroscopic scale in terms of classical thermodynamics; however, the mechanisms explaining thermoelectricity on a microscopic scale are only partially provided by quantum mechanics. Further consideration of the thermoelectric effects on a microscopic scale is beyond the scope of this thesis. By considering both the Peltier and Thomson thermoelectric effects, the thermodynamics of thermoelectricity give rise to the Seebeck effect. The three thermoelectric coefficients defined in the previous sections are not independent but rather are related. Pollock [1982] used the principle of conservation of energy of the thermoelectric circuit system to derive the relationship between the three reversible thermoelectric coefficients.

2.1.4.1 Conservation of Energy

In this section, a reversible thermocouple is considered. It has a temperature difference between its junctions of ΔT; the cooler junction is at temperature T and the warmer junction is at temperature $T + \Delta T$. The emf generated in the thermoelectric circuit is E_{ab} and is given by

$$E_{ab} = \frac{dE_{ab}}{dT} \Delta T \,. \tag{2.6}$$

Conservation of energy of the thermoelectric circuit requires that the Peltier and Thomson thermal effects be included. The thermal and electrical energy balance can be expressed

$$I \left[\frac{dE_{ab}}{dT} \Delta T = \pi_{ab}(T + \Delta T) - \pi_{ab}(T) + \tau_b \, \Delta T - \tau_a \, \Delta T \right] \tag{2.7}$$

where $\pi_{ab}(T + \Delta T)\, I$ is the heat absorbed at the hotter junction, $-\pi_{ab}(T)\, I$ is the heat liberated at the colder junction, $\tau_b \, \Delta T \, I$ is the heat absorbed through conductor *b*, and $-\tau_a \, \Delta T \, I$ is the heat liberated through conductor *a*.

When dividing both sides of Equation 2.7 by ΔT and taking the limit as ΔT approaches zero, the fundamental *theorem of thermoelectricity* is finally obtained,

$$\frac{dE_{ab}}{dT} = \frac{d\pi_{ab}}{dT} + (\tau_b - \tau_a) \,. \tag{2.8}$$

This theorem states that the electrical Seebeck effect is the algebraic sum of the thermal Peltier and Thomson effects.

2.2 Heat Conduction Analysis for a One-Dimensional Composite Medium

The thermal radiation detector being modeled is complex, and it is therefore difficult to formulate a mathematical model that accurately represents its dynamic behavior. Our approach has been to formulate a numerical model of a prototype detector that can be described as a one-dimensional composite medium.

In this section we consider the solution of the steady-state and transient temperature distributions of the one-dimensional composite medium illustrated in Figure 2.5. The adjective "one-dimensional" indicates that only one coordinate is used to describe the spatial variation of temperature. The adjective "steady-state" describes the condition where a system is independent of time. That is, the temperature gradients that exist along the system do not change over time. The adjective "transient" describes the condition where the system is dependent on time in a nonperiodic way and therefore temporal changes of temperature gradient along the system are expected. The adjective "composite" refers to a complex medium in which two or more homogeneous plates are stacked to produce a distinct overall property generally different from those of each of the component plates.

The steady-state and transient temperature distribution analytical solutions are derived for each layer of the one-dimensional composite medium. While the steady-state temperature distribution solution is relatively easy to obtain, the transient solution of the problem is more complicated. The transient temperature distribution analysis in a composite medium requires application of the principle of superposition, the transformation from non-homogeneous boundary conditions to homogeneous boundary conditions, and the orthogonal expansion technique. The solution of the transient temperature distribution $T_i(x,t)$ for any layer i begins by splitting the problem in two parts: the steady-state part $T_{s,i}(x)$ and the transient part $T_{t,i}(x)$, that is

$$T_i(x,t) = T_{s,i}(x) + T_{t,i}(x,t) \,. \qquad (\ 2.9\ )$$

The solution of each part is derived in the following sections.

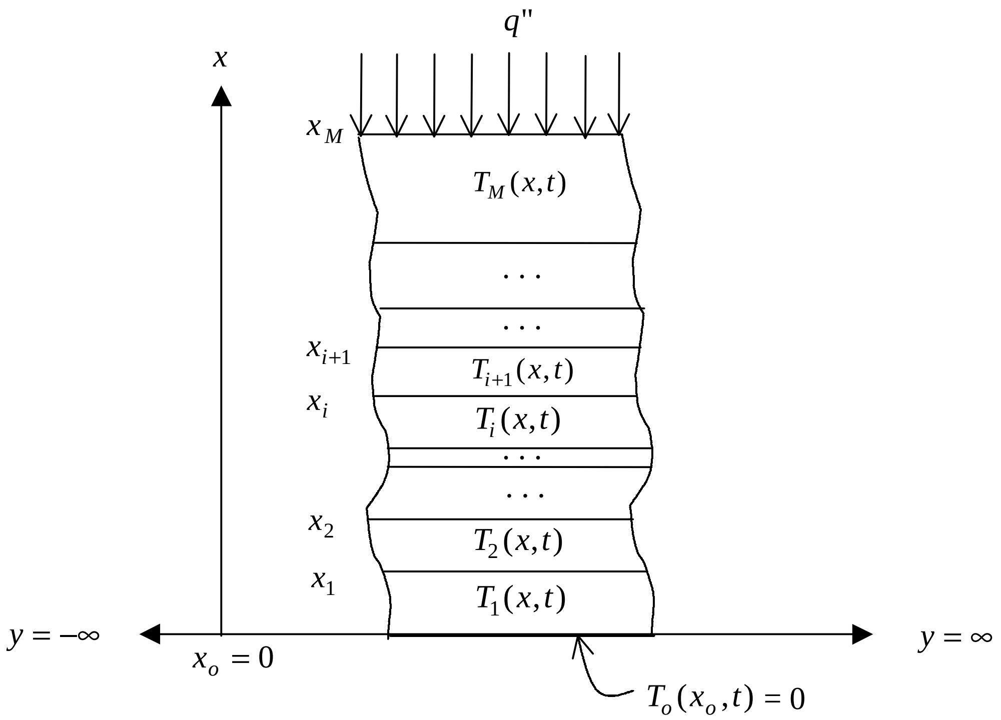


Figure 2.5 Composite medium of *M* parallel plates.

2.2.1 Steady-State Part of the Transient Solution

The steady-state part of the transient solution consists simply of a one-dimensional steady-state conduction solution. Because the steady-state part can be solved easily compared to the transient part, the non-homogeneous boundary condition *q*" is included with the steady-state part, as shown in Figure 2.6. That is,

$$q''(x_m) = q_s''(x_m) + 0. \tag{2.10}$$

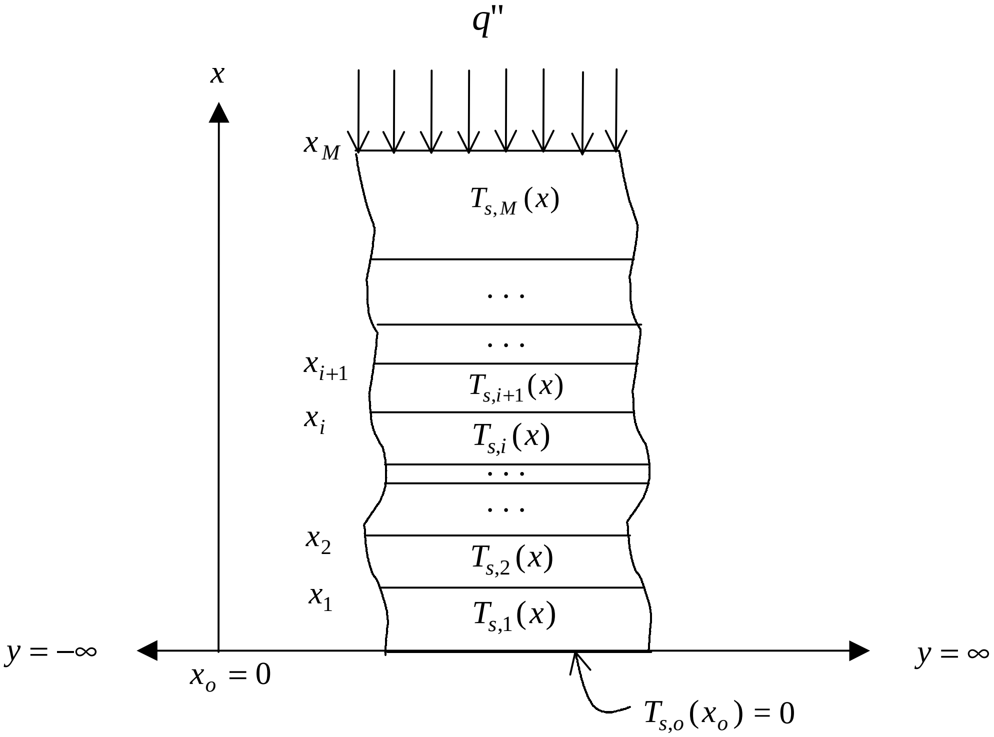


Figure 2.6 Geometry and boundary conditions for the steady-state part of the transient solution.

The equation of steady-state heat conduction for all layers is

$$\frac{\partial^2 T_{s,i}}{\partial x^2} = 0 \quad \text{in} \quad x_{i-1} \le x \le x_i\,, \qquad \text{for } i = 1,2,\ldots,M, \qquad (\ 2.11\ )$$

subject to the boundary conditions

$$T_{s,1}(x_o) = 0\,, \qquad \text{at the bottom,} \qquad (\ 2.12\ )$$

$$T_{s,i}(x_i) = T_{s,i+1}(x_i), \qquad (2.13)$$

$$k_i \left(\frac{\partial T_{s,i}}{\partial x} \right)_{x=x_i} = k_{i+1} \left(\frac{\partial T_{s,i+1}}{\partial x} \right)_{x=x_i}, \qquad (2.14)$$

at the interfaces $i = 1,2,\ldots, M$,

and

$$k_M \left(\frac{\partial T_{s,M}}{\partial x} \right)_{x=x_M} = q'' \qquad (2.15)$$

at the top.

By considering the thermal resistance of each layer i to be analogous to an electrical resistance, Weckmann [1997] formulated the steady-state temperature distributions $T_{s,1}(x)$, $T_{s,2}(x)$, $T_{s,3}(x)$, and $T_{s,4}(x)$. Based on these results, the general solution for each layer is

$$T_{s,i}(x) = q''\left[\frac{x}{k_i} + \sum_{j=1}^{i} x_{j-1}\left(\frac{1}{k_{j-1}} - \frac{1}{k_j} \right) \right]. \qquad (2.16)$$

2.2.2 Transient part of the Transient Solution

The *orthogonal expansion technique* is used to solve for the transient temperature distribution $T_{t,i}$ of layer i in a composite medium of parallel layers of plates. Ozisik's [1993] derivation and solution of this technique was adopted and adapted to solve the homogeneous problem of heat conduction in the composite medium presented in Figure 2.7. Note that the non-homogeneous boundary conditions are included with the steady-state part and therefore the boundary conditions in the transient part are homogeneous.

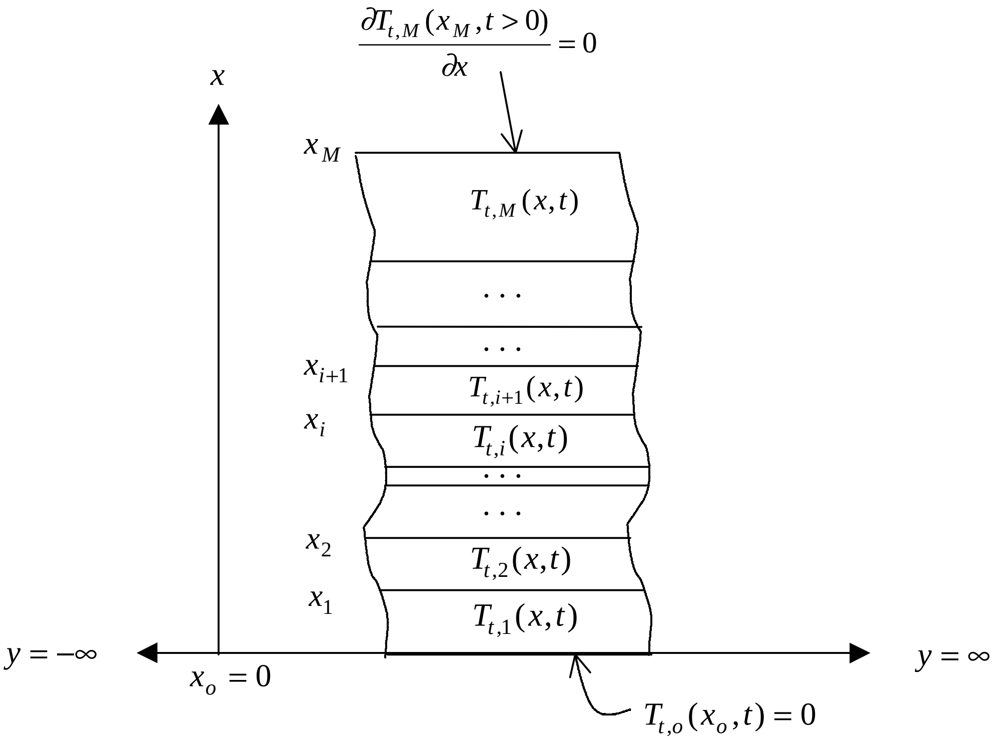


Figure 2.7 Geometry and boundary conditions for the transient part of the transient solution.

The diffusion equation of heat conduction in a composite medium with *M* infinite parallel plates, no temperature gradients along the y and z coordinates, no heat generation, and constant thermophysical properties, as shown in Figure 2.7, is given by

$$\alpha_i \frac{\partial^2 T_{t,i}}{\partial x^2} = \frac{\partial T_{t,i}(x,t)}{\partial t} \qquad \text{in } x_i < x < x_{i+1}, \text{ for } t = 0, i = 1,2,\ldots, M. \qquad (2.17)$$

subject to the boundary conditions

$$T_{t,i} = 0, \qquad \text{at } x = 0, \text{ for } t > 0, \qquad (2.18)$$

$$\left.\begin{aligned} &T_{t,i} = T_{t,i+1}, && (2.19)\\ &k_i \frac{\partial T_{t,i}}{\partial x} = k_{i+1}\frac{\partial T_{t,i+1}}{\partial x}, && (2.20)\end{aligned}\right\} \text{ at interfaces } x = x_i,\ \text{for } t > 0$$

$$k_M \frac{\partial T_{t,M}}{\partial x} = 0, \qquad \text{at } x = x_M, \text{ for } t > 0, \qquad (2.21)$$

and the initial conditions

$$T_{t,i}(x,t) = f_i(x) \qquad \text{in } x_i < x < x_{i+1}, \text{ for } t = 0, i = 1,2,\ldots,M. \qquad (2.22)$$

In order to solve for the transient part $T_{t,i}(x,t)$, in layers i =1,2,…, M of a composite medium, the *separation of variables* technique is introduced in the form

$$T_{t,i}(x,t) = \psi_i(x)\,\Gamma(t), \qquad (2.23)$$

where $\psi_i(x)$ is a space-dependent function and $\Gamma(t)$ is a time-dependent function. By introducing $T_{t,i}(x,t)$ into the mathematical formulation and dividing each term of the resulting expression by $\psi_i(x)\,\Gamma(t)$, these results

$$\alpha_i \frac{1}{\psi_i(x)}\frac{d^2\psi_i}{dx^2} = \frac{1}{\Gamma(t)}\frac{d\Gamma}{dt} = -\beta^2. \qquad (2.24)$$

In Equation 2.23 β is the so-called separation constant. The left-hand side of Equation 2.23 is independent of time, and the right-hand side is independent of x and α_i. The two sides can be equal to each other only if they are both equal to the same negative constant β^2.

The resulting ordinary differential equation corresponding to the time-dependent function $\Gamma(t)$ and its corresponding solution are given by

$$\frac{d\Gamma(t)}{dt} + \beta_n^2 \Gamma(t) = 0 \qquad \text{for } t > 0, \qquad (2.25)$$

and

$$\Gamma(t) = e^{-\beta_n^2 t}, \qquad \text{respectively.} \qquad (2.26)$$

The determination of the space-dependent function $\psi_i(\beta, x)$ requires the solution of the *eigenvalue problem*. This problem is described by the ordinary differential equation

$$\frac{d^2\psi_{in}}{dx^2} + \frac{\beta_n^2}{\alpha_i}\psi_{in} = 0 \qquad \begin{array}{l} \text{in } x_i < x < x_{i+1}, \\ i = 1,2,\ldots, M, \end{array} \qquad (2.27)$$

and the corresponding boundaries conditions

$$\psi_{in} = 0 \qquad \text{at } x = 0, \qquad (2.28)$$

$$\left.\begin{array}{r} \psi_{in} = \psi_{i+1,n} \qquad (2.29) \\ \\ k_i \dfrac{\partial \psi_{in}}{\partial x} = k_{i+1}\dfrac{\partial \psi_{i+1,n}}{\partial x} \qquad (2.30) \end{array}\right\} \quad \begin{array}{l} \text{at the interfaces } x = x_i, \\ \text{where } i = 1,2,\ldots, M-1, \end{array}$$

$$k_M \frac{\partial \psi_{Mn}}{\partial x} = 0 \qquad \text{at } x = x_M, \text{ for } t > 0, \qquad (2.31)$$

where $\psi_i(\beta_n, x)$ is simplified as ψ_{in} and the subscript n indicates that an infinite number of discrete values of β exists for the corresponding solution of the eigenfunction ψ_i. Note that the boundary conditions given in the eigenvalue problem were obtained by introducing $T_{t,i}(x,t) = \psi_i(x)\,\Gamma(t)$ into the boundary conditions for $T_{t,i}(x,t)$. The general solution of $\psi_{in}(x)$ can be written

$$\psi_{in}(x) = A_{in}\sin\left(\frac{\beta_n}{\sqrt{\alpha_i}}x\right) + B_{in}\cos\left(\frac{\beta_n}{\sqrt{\alpha_i}}x\right) \qquad \text{in } x_{i-1} < x < x_i\,,\ \text{for } i = 1,2,\ldots M, \qquad (2.32)$$

where A_{in} and B_{in} are the integration constants.

Introducing these general equations into the boundary conditions identified in the eigenvalue problem, there results

$$A_{1n}\sin(\eta_1) + B_{1n}\cos(\eta_1) = 0\,, \qquad (2.33)$$

$$A_{in}\sin(\eta_i) + B_{in}\cos(\eta_i) = A_{i+1,n}\sin\left(\frac{x_i}{x_{i+1}}\eta_{i+1}\right) + B_{i+1,n}\cos\left(\frac{x_i}{x_{i+1}}\eta_{i+1}\right),$$

$$\text{for } i = 1,2,\ldots, M - 1, \qquad (2.34)$$

$$K_{i,i+1}\{A_{in}\cos(\eta_i) - B_{in}\sin(\eta_i)\} = A_{i+1,n}\cos\left(\frac{x_i}{x_{i+1}}\eta_{i+1}\right) - B_{i+1,n}\sin\left(\frac{x_i}{x_{i+1}}\eta_{i+1}\right),$$

$$\text{for } i = 1,2,\ldots, M - 1, \qquad (2.35)$$

and

$$A_{Mn}\cos(\eta_M) - B_{Mn}\sin(\eta_M) = 0\,, \qquad (2.36)$$

where $\eta_i = \dfrac{\beta_n}{\sqrt{\alpha_i}}x_i$ and $K_{i,i+1} = \dfrac{k_i}{k_{i+1}}\sqrt{\dfrac{\alpha_{i+1}}{\alpha_i}}$.

For $x = 0$, note that the equation $A_{1n}\sin(\eta_1) + B_{1n}\cos(\eta_1) = 0$ requires that $B_{1n} = 0$.

The above equations can be expressed in the matrix form

$$\begin{bmatrix} \sin(\eta_1) & -\sin\left(\frac{x_1}{x_2}\eta_2\right) & -\cos\left(\frac{x_1}{x_2}\eta_2\right) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ K_{12}\cos(\eta_1) & -\cos\left(\frac{x_1}{x_2}\eta_2\right) & \sin\left(\frac{x_1}{x_2}\eta_2\right) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \sin(\eta_2) & \cos(\eta_2) & -\sin\left(\frac{x_2}{x_3}\eta_3\right) & -\cos\left(\frac{x_2}{x_3}\eta_3\right) & \dots & 0 & 0 & 0 & 0 \\ 0 & K_{23}\cos(\eta_2) & \text{-}K_{23}\sin(\eta_2) & -\cos\left(\frac{x_2}{x_3}\eta_3\right) & \sin\left(\frac{x_2}{x_3}\eta_3\right) & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & \sin(\eta_{M-1}) & \cos(\eta_{M-1}) & -\sin\left(\frac{x_{M-1}}{x_M}\eta_M\right) & -\cos\left(\frac{x_{M-1}}{x_M}\eta_M\right) \\ 0 & 0 & 0 & 0 & 0 & \dots & K_{M-1,M}\cos(\eta_{M-1}) & \text{-}K_{M-1,M}\sin(\eta_{M-1}) & -\cos\left(\frac{x_{M-1}}{x_M}\eta_M\right) & \sin\left(\frac{x_{M-1}}{x_M}\eta_M\right) \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \cos(\eta_M) & \sin(\eta_M) \end{bmatrix} \begin{bmatrix} A_{1,n} \\ A_{2,n} \\ B_{2,n} \\ A_{3,n} \\ \dots \\ B_{M-1,n} \\ A_{M,n} \\ B_{M,n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.37}$$

The above matrix form of Equation 2.36 can be recognized as a homogeneous linear system in the form $\xi\,\mathbf{x} = \mathbf{0}$, where ξ is a square matrix and $\mathbf{0}$ is a zero vector. Solving the transcendental equation $\xi = 0$ yields the *eigenvalues* β_n.

The constants A_{in} and B_{in}, i =2,3,…, M, are determined by solving

$$\begin{bmatrix} \sin\left(\frac{x_{i-1}}{x_i}\eta_i\right) & \cos\left(\frac{x_{i-1}}{x_i}\eta_i\right) \\ \cos\left(\frac{x_{i-1}}{x_i}\eta_i\right) & -\sin\left(\frac{x_{i-1}}{x_i}\eta_i\right) \end{bmatrix} \begin{bmatrix} A_{in} \\ B_{in} \end{bmatrix} = \begin{bmatrix} A_{i-1,n}\sin(\eta_{i-1}) + B_{i-1,n}\cos(\eta_{i-1}) \\ K_{i-1,i}\left\{A_{i-1}\cos(\eta_{i-1}) - B_{i-1,n}\sin(\eta_{i-1})\right\} \end{bmatrix}, \tag{2.38}$$

using Cramer's rule. When this is done there results

$$A_{in} = \frac{1}{\delta_{in}}\left[-\left\{A_{i-1,n}\sin(\eta_{i-1}) + B_{i-1,n}\cos(\eta_{i-1})\right\}\sin\left(\frac{x_{i-1}}{x_i}\eta_i\right)\right.$$
$$\left. -K_{i-1,i}\left\{A_{i-1,n}\cos(\eta_{i-1}) + B_{i-1,n}\sin(\eta_{i-1})\right\}\cos\left(\frac{x_{i-1}}{x_i}\eta_i\right)\right] \tag{2.39}$$

and

$$B_{in} = \frac{1}{\delta_{in}}\left[K_{i-1,i}\left\{A_{i-1,n}\cos(\eta_{i-1}) - B_{i-1,n}\sin(\eta_{i-1})\right\}\sin\left(\frac{x_{i-1}}{x_i}\eta_i\right)\right.$$

$$\left. -\left\{A_{i-1,n}\sin(\eta_{i-1}) + B_{i-1,n}\cos(\eta_{i-1})\right\}\cos\left(\frac{x_{i-1}}{x_i}\eta_i\right)\right], \qquad (\ 2.40\ )$$

where

$$-\delta_{in} = \sin^2\left(\frac{x_{i-1}}{x_i}\eta_i\right) + \cos^2\left(\frac{x_{i-1}}{x_i}\eta_i\right) = 1. \qquad (\ 2.41\ )$$

Now, the general solution of the transient temperature distribution $T_{t,i}(x,t)$, in layers i =1,2,…, M , can be expressed as

$$T_{t,i}(x,t) = \sum_{n=1}^{\infty} C_n e^{-\beta_n^2 t}\psi_{in}(x). \qquad (\ 2.42\ )$$

The coefficient C_n can be evaluated by introducing this general solution of $T_i(x,t)$ into the previously stated initial condition, $T_{t,i}(x,t=0) = f_i(x)$. The result is the expression of $f_i(x)$ in a Fourier series expansion

$$f_i(x) = \sum_{n=1}^{\infty} C_n \psi_{in}(x). \qquad (\ 2.43)$$

The unknown coefficient C_n can be evaluated by multiplying both sides by $\frac{k_i}{\alpha_i}\psi_{ik}(x)$, integrating the resulting expression over the interval (x_{i-1}, x_i), and summing the resulting integration over all regions, leading to

$$\sum_{i=1}^{M}\frac{k_i}{\alpha_i}\int_{x_{i-1}}^{x_i}\psi_{ik}(x)f_i(x)\ dx=\sum_{i=1}^{M}C_n\left[\sum_{i=1}^{M}\frac{k_i}{\alpha_i}\int_{x_{i-1}}^{x_i}\psi_{ik}(x)\psi_{in}(x)\ dx\right]. \tag{2.44}$$

Mikhailov and Ozisik [1994] identify the orthogonality condition

$$\sum_{i=1}^{M}\frac{k_i}{\alpha_i}\int_{x_{i-1}}^{x_i}\psi_{ik}(x)\psi_{in}(x)\ dx=\begin{cases}0 & \text{for} \quad n\neq k\\ N_n & \text{for} \quad n=k\end{cases}, \tag{2.45}$$

where $\psi_{ik}(x)$ and $\psi_{in}(x)$ are two different *eigenfunctions* and N_n is the normalization integral of the function $\psi_{in}(x)$ defined as

$$N_n=\sum_{i=1}^{M}\frac{k_i}{\alpha_i}\int_{x_{i-1}}^{x_i}\psi_{in}^2(x)\ dx. \tag{2.46}$$

Based on this orthogonality condition, the coefficients of C_n are obtained as

$$C_n=\frac{1}{N_n}\sum_{i=1}^{M}\int_{x_{i-1}}^{x_i}\psi_{in}(x)f_i(x)\ dx. \tag{2.47}$$

Finally, the solution of the transient temperature distribution of a composite medium with *M* infinite parallel plates, no temperature gradients along the *y* and *z* coordinates, and constant thermophysical properties, as shown in Figure 2.6, is given by

$$T_{t,i}(x,t)=\sum_{n=1}^{\infty}\frac{e^{-\beta_n^2 t}\psi_{in}(x)}{N_n}\sum_{i'=1}^{M}\frac{k_{i'}}{\alpha_{i'}}\int_{x_{i'-1}}^{x_{i'}}\psi_{i'n}(x')f_{i'}(x')\ dx'$$

in $x_{i-1}<x<x_i$, for t >0, for layers i = 1,2,…,*M*. (2.48)

The thermoelectricity concepts introduced in Section 2.1 and the thermal analysis introduced in Section 2.2 are extensively used in the following chapters. In Chapter 3, the experimental design, developed to estimate the electro-thermophysical properties of the thermal radiation detector, is introduced. The term "experimental design" is used to describe the following steps: (1) development of an engineering prototype detector concept, (2) development of an analytical model of the engineering prototype detector concept, and (3) the implementation of a parameter estimation procedure. In Chapter 4, numerical simulations of experiments provide insight into how the accuracy of the measurement system and the nature of the experiments can influence the estimation of the electro-thermophysical properties.

3. Experimental Design

The eventual development of a dynamic electrothermal model to accurately simulate the new detector for Earth radiation budget applications requires reliable estimates of the electro-thermophysical properties of the materials used. However, the available literature lacks reliable information about the thermophysical properties. In addition, alterations to these properties as a consequence of thin-film manufacturing technology, including the effect of contact resistance, are possible and are not well understood.

Accurate property estimates can be obtained by the careful design of experiments that utilize modeling in combination with experimental measurements on prototype detectors. The experimental design carried out in this investigation involves the following processes: (1) the design of an engineering prototype detector (Section 3.1), (2) the mathematical model of the engineering prototype (Section 3.2), and (3) the implementation of a parameter estimation procedure (Section 3.3).

The prototype detector is an operational detector that closely approximates the actual dimensional specifications and operation of the thin-film thermocouple thermal radiation concept. The prototype detector would be sufficiently similar to the actual detector so that the measured material properties would be the same for both.

The mathematical model is developed to simulate the temperature distribution within the experimental item, a multi-layer composite thermocouple detector. The heat conduction analysis is simplified by the application of acceptable assumptions that do not compromise the ability of the model to accurately represent the detector behavior.

Once the analytical model has been developed, the experimental design is finally characterized by introducing a parameter estimation procedure. This procedure combines both theoretical and experimental results to obtain reliable estimates of the electrothermal properties of the detector.

3.1 Design of the Engineering Prototype Detector

The development of the dynamic electrothermal model to accurately simulate the behavior of a thermal radiation detector requires reliable estimations of the electro-thermophysical properties. In such cases it becomes necessary to gather and analyze data through experimentation and modeling. In the case at hand the detector being modeled is complex, and is difficult to formulate a model that accurately represents its dynamic behavior. Our approach has been to formulate a theoretical model of a simpler prototype detector to be used to measure the electro-thermophysical properties. This prototype detector would incorporate the same design and operating features as the actual detector but would be geometrically simpler to model.

The design objectives of the prototype detector take into account the possibility of developing a heat transfer analysis where the temperature distribution and thermoelectricity effects can be accurately predicted from first principles. Several configurations were considered that may be used to attain the design objectives. The most suitable of these is shown in Figure 3.1.

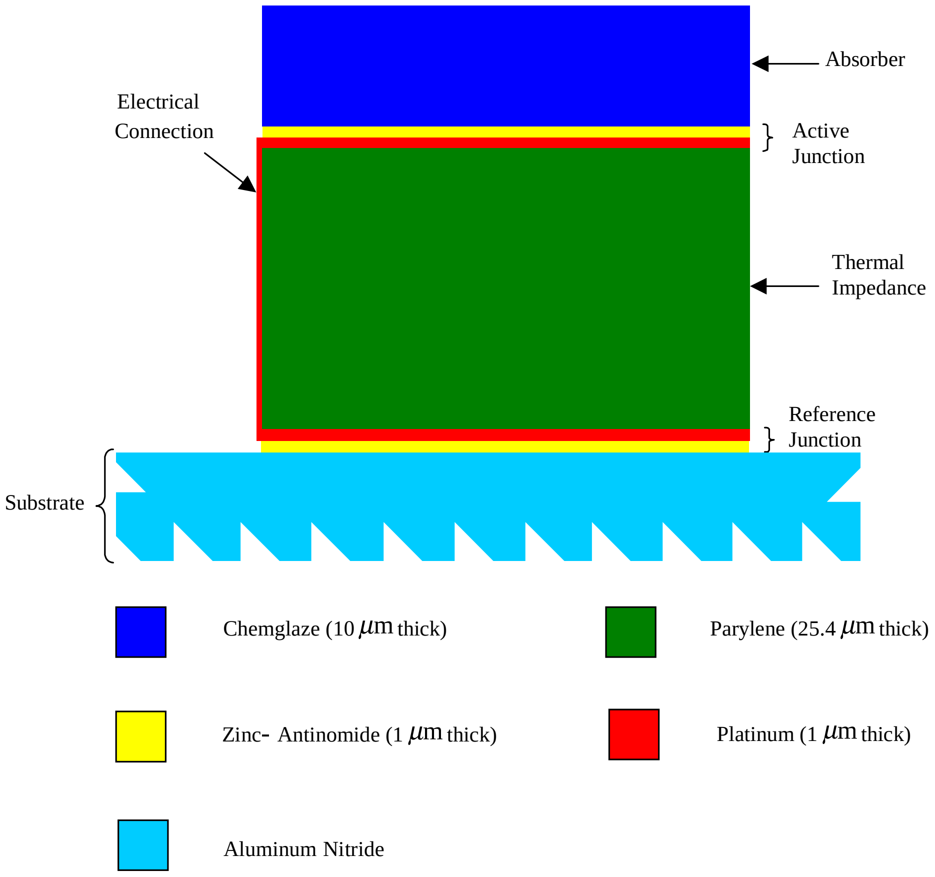

Absorber
Electrical Connection
Active Junction
Thermal Impedance
Reference Junction
Substrate
Chemglaze (10 μm thick)
Parylene (25.4 μm thick)
Zinc- Antinomide (1 μm thick)
Platinum (1 μm thick)
Aluminum Nitride

Figure 3.1 The engineering prototype detector for measuring electro-thermophysical properties (not to scale).

From the manufacturing point of view, the engineering prototype detector closely resembles the actual thermal radiation detector, shown in Figure 1.3. From top to bottom, the engineering prototype detector consists of an absorber layer (Chemglaze), an active junction (sputtered zinc-antinomide and platinum layers), a thermal impedance layer (vapor-deposited parylene), and reference junction (sputtered zinc-antinomide and platinum layers), assembled on top of an aluminum-nitride substrate. As for the case of the actual detector concept, the thin-film manufacturing technique is capable of providing accurate thickness dimensions as well as uniform chemical compositions of the layers. Moreover, fabrication under vacuum conditions assures that the degradation of the performance of the detector does not occur due to outgassing and related debonding. Although the electrical connection between the junctions might prove to be a fabrication challenge, several alterations to the design can be considered while preserving the basic design concept, a one-dimensional composite medium.

The prototype detector operational dynamics closely resembles that of the actual detector. Specifically, a small voltage proportional to the absorbed heat flux is created by the temperature difference across the thermal impedance layer when the device is exposed to thermal radiation. In theory, the engineering prototype can assume the role of the detector in a scheme for recovering the electro-thermophysical properties of the detector materials.

3.2 Mathematical Model of the Engineering Prototype

Several numerical methods are available that could possibly be used to analyze the dynamic thermal response of composite mediums. However, a closed-form analytical approach was followed in our analysis for several reasons. First of all, the analytical approach leads to algebraic expressions for individual temperature distributions in each of the layers of the composite medium. The ability to determine the exact mathematical solutions for the temperature distribution of each layer can be extremely useful, specifically in the case of functional optimization. Second, an analytical solution provides the flexibility to be parameterized and applied for any combination of dimensions and electro-thermophysical properties. Also, in contrast to numerical solutions, which estimate the temperature at discrete locations and use approximating polynomials to estimate the temperature distribution elsewhere, an analytical solution can determine the temperature at any specific location of interest.

The determination of the boundary conditions and other assumptions is presented in the following section. Afterwards, the solutions of the steady-state and transient analytical temperature distributions are presented for each layer of the engineering prototype.

3.2.1 Determination of Boundary Conditions and General Assumptions

Since the actual detector is intended to be deployed in Earth orbit, the analytical model of the engineering prototype assumes the same environmental constraints and, therefore, only two modes of heat transfer are considered in the investigation: radiation and conduction. Moreover, the inputs to the model of the engineering prototype are the same as those encountered in the actual detector. The boundary conditions applied to the analytical model of the engineering prototype are shown in Figure 3.2.

A uniform distributed heat flux ($q'' = 1.0$ W/m^2) is applied at the top of the absorber layer at time $t = 0$. This heat flux represents the incident thermal radiation arriving from the Earth scene through the instrument optics. The absorber layer, on the thermopile active junction, has a high absorptivity and a low reflectivity, and the reflectivity is mostly specular. Thus, it is assumed that all radiation heat flux incident to the absorber layer is completely absorbed. Said another way, after reflection the absorbed flux is 1.0 W/m^2. The heat sink defines a fixed-temperature plane boundary (311 K) at the bottom of the prototype detector. This fixed temperature represents the temperature of the integrated cavity and the heat sink of the actual thermopile detector.

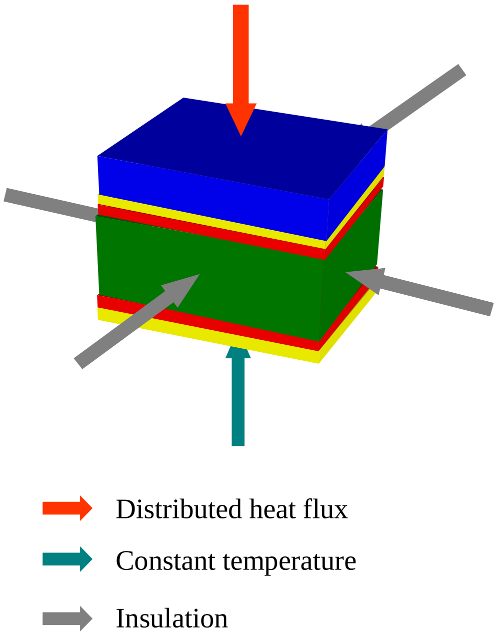


Figure 3.2 Boundary conditions applied to analytical model of the engineering prototype.

One advantage of the model is that its responsivity is inherently dependent on the temperature difference between the active and reference junctions. In order to reduce the complexity of the problem and to focus only on the temperature difference between the junctions, a variable transformation is introduced to transform the fixed temperature boundary condition at $T_o' = 311\,K$, a non-homogeneous boundary condition, to $T_o = T_o' - 311 = 0$, a homogeneous boundary condition. Therefore, the calculated temperature at any location through the prototype detector is really the temperature difference between that point and the substrate, and not the absolute temperature.

The analytical model assumes that the lateral dimensions of the layers are infinite; thus edge effects are neglected. For a similar configuration case, Weckmann [1997] demonstrates through a finite element analysis that two-dimensional heat transfer effects can be ignored in the analysis of the active junction of the thermal radiation detector.

3.2.2 One-Dimensional Analytical Model

As depicted in Figure 3.3, the analytical model can be described as a one-dimensional composite medium consisting of six parallel layers. The steady-state and transient analytical temperature distribution solutions are derived for each layer of the one-dimensional detector in Section 2.2. The general steady-state solution is

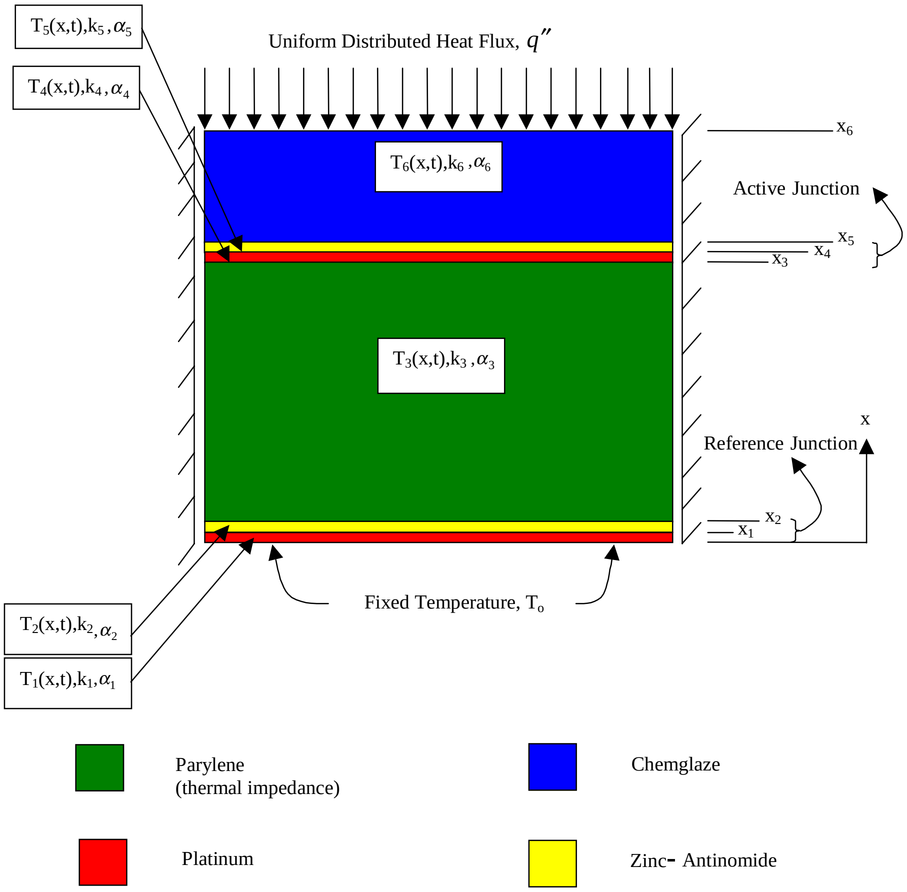
$$T_{s,i}(x) = q''\left[\frac{x}{k_i} + \sum_{j=1}^{i} x_{j-1}\left(\frac{1}{k_{j-1}} - \frac{1}{k_j}\right)\right]$$

for layers i = 1,2,3,4,5,6. (3.1)

The general solution of the transient temperature distribution $T_i(x,t)$ for any layer is

$$T_i(x,t) = T_{s,i}(x) - \sum_{n=1}^{\infty} e^{-\beta_n^2 t} \psi_{in}(x) \frac{\sum_{i'=1}^{6} \frac{k_{i'}}{\alpha_{i'}} \int_{x_{i'-1}}^{x_{i'}} \psi_{i'n}(x') T_{s,i'}(x') \, dx'}{\sum_{i'=1}^{6} \frac{k_{i'}}{\alpha_{i'}} \int_{x'_{i'-1}}^{x'_{i'}} \psi_{i'n}^2(x') \; dx'}$$

in $x_{i-1} < x < x_i$, for $t > 0$ and layers $i = 1,2,3,4,5,6$. (3.2)



$T_5(x,t), k_5, \alpha_5$
$T_4(x,t), k_4, \alpha_4$
Uniform Distributed Heat Flux, q''
$T_6(x,t), k_6, \alpha_6$
x_6
Active Junction
x_5
x_4
x_3
$T_3(x,t), k_3, \alpha_3$
x
Reference Junction
x_2
x_1
Fixed Temperature, T_o
$T_2(x,t), k_2, \alpha_2$
$T_1(x,t), k_1, \alpha_1$
Parylene (thermal impedance)
Chemglaze
Platinum
Zinc- Antinomide

Figure 3.3 Thermal model of prototype detector (not to scale).

The space-dependent function ψ_{in}, i =1,2,3,4,5,6, is given by

$$\psi_{in}(x) = A_{in} \sin\left(\frac{\beta_n}{\sqrt{\alpha_i}} x \right) + B_{in} \cos\left(\frac{\beta_n}{\sqrt{\alpha_i}} x \right) \qquad \text{in} \quad x_{i-1} < x < x_i \,. \quad (3.3)$$

With $B_{1n} = 0$ and $A_{1n} = 1$, the constants A_{in} and B_{in}, i =2,3,4,5,6, are determined as

$$\begin{aligned} A_{in} = -\Bigg[& -\left\{ A_{i-1,n} \sin(\eta_{i-1}) + B_{i-1,n} \cos(\eta_{i-1}) \right\} \sin\left(\frac{x_{i-1}}{x_i} \eta_i \right) \\ & - K_{i-1,i} \left\{ A_{i-1,n} \cos(\eta_{i-1}) + B_{i-1,n} \sin(\eta_{i-1}) \right\} \cos\left(\frac{x_{i-1}}{x_i} \eta_i \right) \Bigg] \end{aligned} \qquad (3.4)$$

and

$$\begin{aligned} B_{in} = -\Bigg[& K_{i-1,i} \left\{ A_{i-1,n} \cos(\eta_{i-1}) - B_{i-1,n} \sin(\eta_{i-1}) \right\} \sin\left(\frac{x_{i-1}}{x_i} \eta_i \right) \\ & - \left\{ A_{i-1,n} \sin(\eta_{i-1}) + B_{i-1,n} \cos(\eta_{i-1}) \right\} \cos\left(\frac{x_{i-1}}{x_i} \eta_i \right) \Bigg]. \end{aligned} \qquad (3.5)$$

The eigenvalues, β_n, are determined by finding the roots of the transcendental equation

$$\begin{bmatrix} \sin(\eta_1) & -\sin\left(\frac{x_1}{x_2}\eta_2\right) & -\cos\left(\frac{x_1}{x_2}\eta_2\right) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K_{12}\cos(\eta_1) & -\cos\left(\frac{x_1}{x_2}\eta_2\right) & \sin\left(\frac{x_1}{x_2}\eta_2\right) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sin(\eta_2) & \cos(\eta_2) & -\sin\left(\frac{x_2}{x_3}\eta_3\right) & -\cos\left(\frac{x_2}{x_3}\eta_3\right) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{23}\cos(\eta_2) & -K_{23}\sin(\eta_2) & -\cos\left(\frac{x_2}{x_3}\eta_3\right) & \sin\left(\frac{x_2}{x_3}\eta_3\right) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\eta_3) & \cos(\eta_3) & -\sin\left(\frac{x_3}{x_4}\eta_4\right) & -\cos\left(\frac{x_3}{x_4}\eta_4\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{34}\cos(\eta_3) & -K_{34}\sin(\eta_3) & -\cos\left(\frac{x_3}{x_4}\eta_4\right) & \sin\left(\frac{x_3}{x_4}\eta_4\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin(\eta_4) & \cos(\eta_4) & -\sin\left(\frac{x_4}{x_5}\eta_5\right) & -\cos\left(\frac{x_4}{x_5}\eta_5\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{45}\cos(\eta_4) & -K_{45}\sin(\eta_4) & -\cos\left(\frac{x_4}{x_5}\eta_5\right) & \sin\left(\frac{x_4}{x_5}\eta_5\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sin(\eta_5) & \cos(\eta_5) & -\sin\left(\frac{x_5}{x_6}\eta_6\right) & -\cos\left(\frac{x_5}{x_6}\eta_6\right) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{56}\cos(\eta_5) & -K_{56}\sin(\eta_5) & -\cos\left(\frac{x_5}{x_6}\eta_6\right) & \sin\left(\frac{x_5}{x_6}\eta_6\right) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cos(\eta_6) & -\sin(\eta_6) \end{bmatrix} = 0$$

$$(3.6)$$

where $\eta_i = \dfrac{\beta_n}{\sqrt{\alpha_i}} x_i$ and $K_{i,i+1} = \dfrac{k_i}{k_{i+1}} \sqrt{\dfrac{\alpha_{i+1}}{\alpha_i}}$.

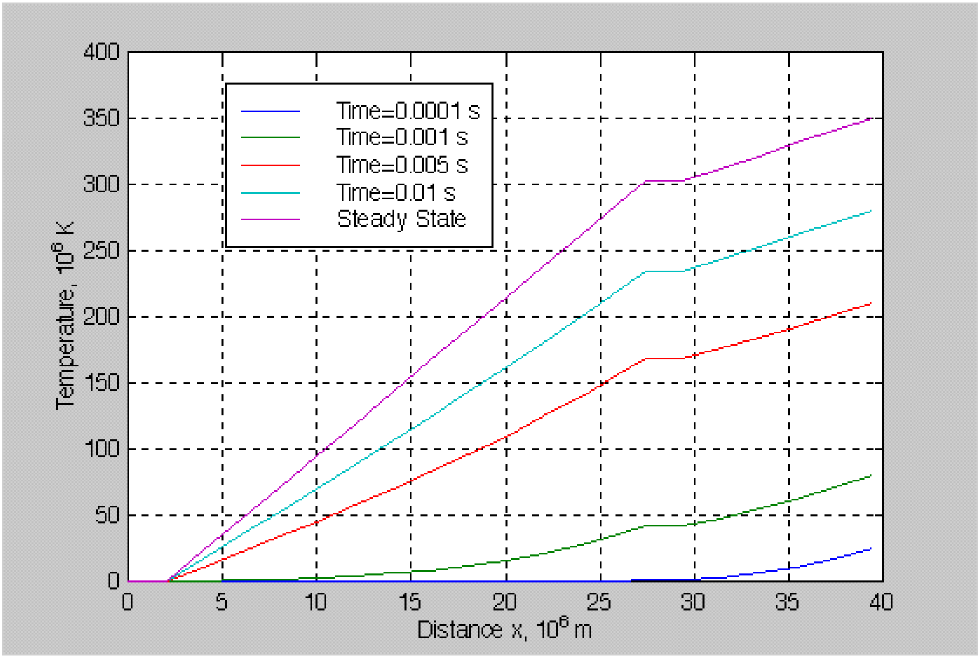


Figure 3.4 Transient temperature distribution of analytical model.

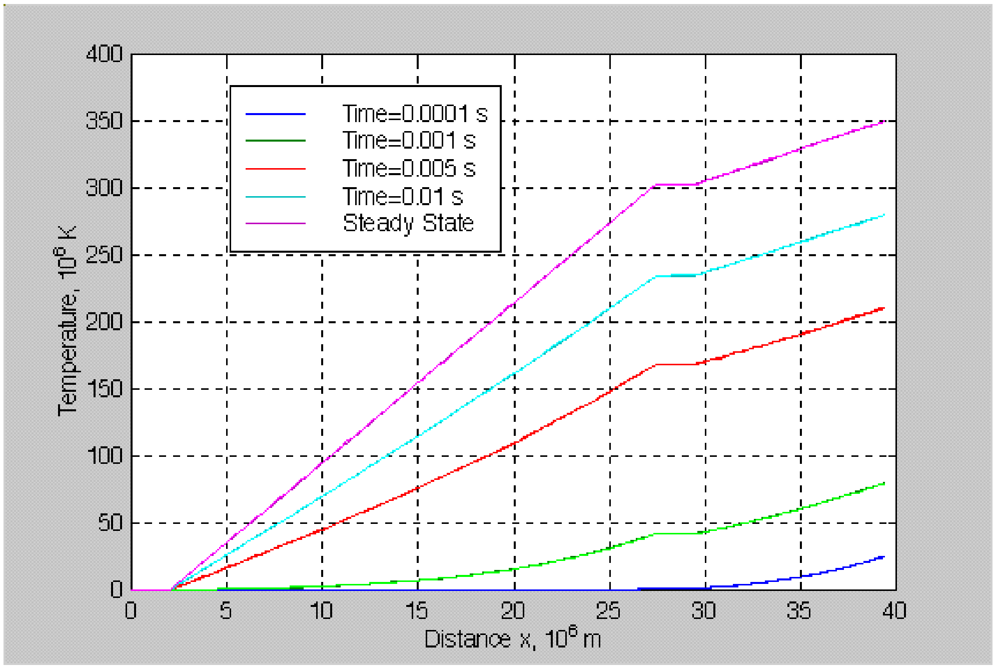


Figure 3.5 Transient temperature distribution of finite element model.

The analytical solution, shown in Figure 3.4, is in agreement with an ALGOR (finite element method) solution, shown in Figure 3.5. The difference between the analytical and the finite element models is less than one percent. In these simulations, the boundary conditions are as described in the previous section, and the initial temperature through the six layers is the same as the heat sink temperature.

3.3 Parameter Estimation

Parameter estimation is the process of searching for unknown values within mathematical models. For example, parameter estimation can be used to determine the unknown thermal properties of a heat conduction model whose only available information is its temperature distribution. The type of problem in which the main goal is to find the parameters that minimize a given function is called a parameter estimation problem. In the previous example, the parameter estimation problem consists of minimizing a function so that unknown thermal transport properties of a heat conduction model can be recovered.

The estimation of properties requires both experimental measurements and an accurate mathematical model of the system whose properties are being determined. The idea is to fit the theoretical model response to the experimental data. The best approach is to consider a function describing the deviation of the theoretical model response from the experimental data. Then a minimization method can be used to determine the parameters so that deviations of the theoretical model response and the experimental data are minimal.

The primary objective of this section is to describe how a parameter estimation procedure, illustrated in Figure 3.6, can be used to estimate the electro-thermophysical properties of the materials making up the thermal radiation detector. The boundary and initial conditions, and the geometry descriptions are specified in the mathematical model of the prototype detector in order to calculate the theoretical temperature. Then a minimization method can be used to determine the material properties of the prototype detector so that deviations of the calculated temperatures and the experimental data are minimal.

In the following sections, the considerations for determining the electro-thermophysical properties of the detector are explained separately. Then the minimization method used to determine the electro-thermophysical properties is introduced.

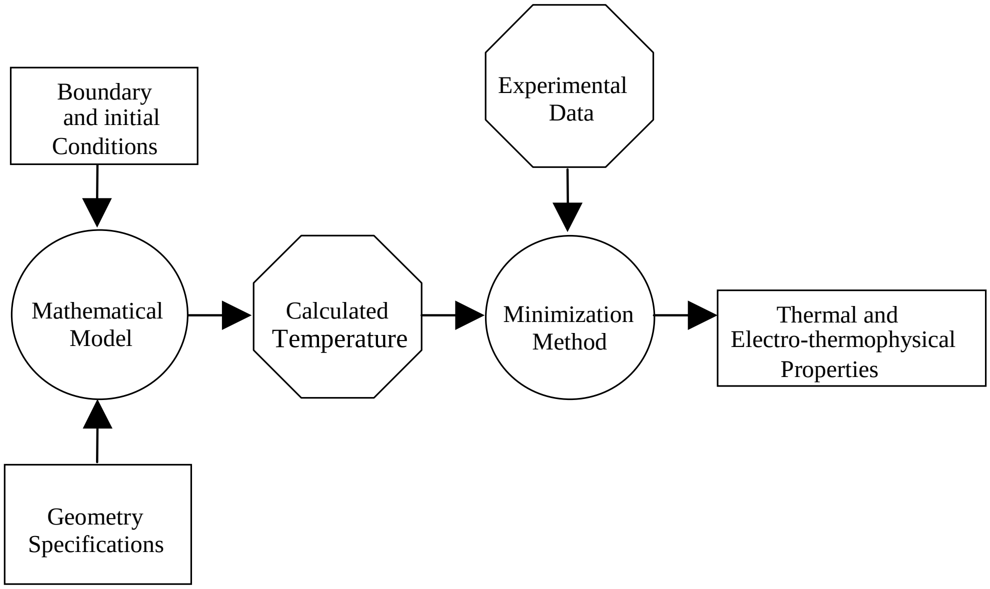


Figure 3.6 Parameter estimation logic [Scott, 1997].

3.3.1 Considerations for Determining the Thermal Properties

The boundary conditions applied to the analytical model of the engineering prototype reproduce as nearly as possible the environmental conditions of the actual thermal radiation detector. The resulting analytical model can provide a thorough understanding of the dynamic electro-thermal behavior of the system under realistic conditions. Specifically, the simulations of the dynamic electrothermal behavior of the engineering prototype, reported elsewhere in this thesis, indicate that extremely small temperature excursions ($350\times10^{-6}\,\mathrm{K}$) will be encountered in the actual detector. As a result it can be concluded that changes of the transport property values due to temperature variations in both the engineering prototype and the actual device are extremely small and can be neglected. Hence, the thermal conductivity and diffusivity of the materials used to fabricate the experimental model are assumed to be constant.

The strategy now is to determine which properties can be estimated indirectly through a combination of modeling and experimental measurements on the engineering prototype detector. Ideally, multiple temperature measurements would be conducted in several locations throughout the prototype. However, measurement constraints, due to the small size of the detector, restrict our ability to obtain several temperature measurements in each of the layers and, therefore, to estimate explicitly all properties. In the current study, only three locations are considered for temperature measurements: (1) at the top, (2) at the substrate surface and (3) at the active junction of the prototype detector (refer to Figure 3.1). Several methods can be utilized to measure the temperature at these locations:

a) A thermographic method can be used to measure the temperature at top of the absorber. The thermographic method has the advantage that, being non-contact and non-invasive, it is capable of providing accurate readings, or measurements, during transient operations.

b) The temperature at the active and reference junctions of the prototype detector can be measured by building a resistance temperature sensor (RTS) along the platinum layer, as shown in Figure 3.7. An advantage of this method is the accurate time response of the RTS.

c) The temperature difference between the active and reference junctions of the prototype detector can be determined by measuring the voltage difference between the junctions, e.g. calibration.

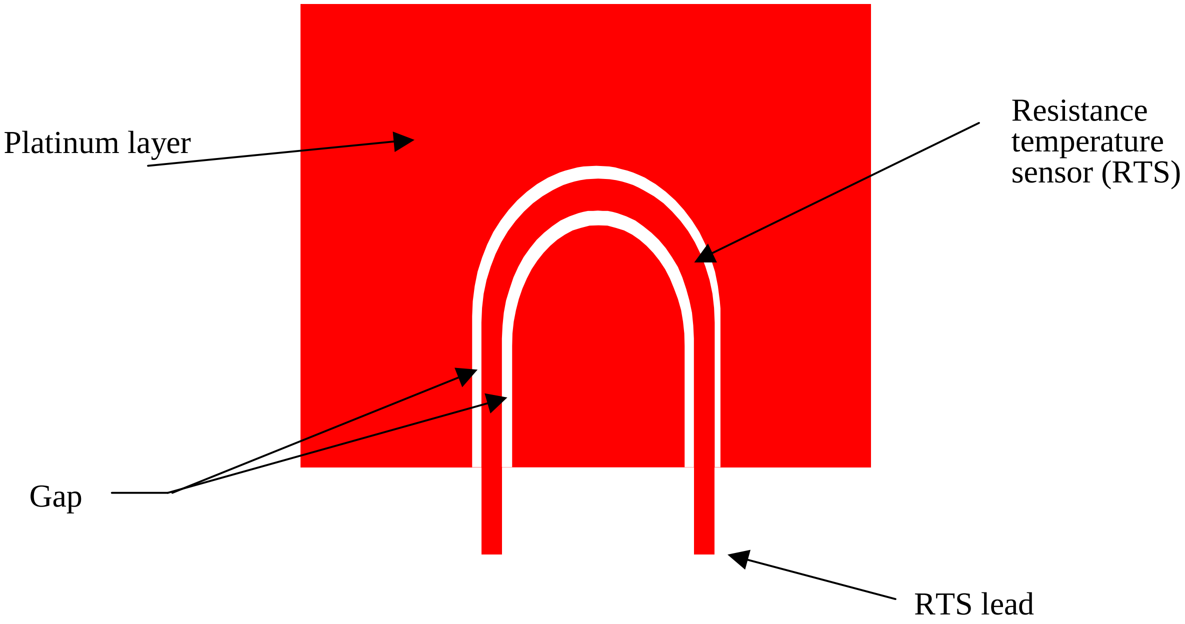


Figure 3.7 Resistance temperature sensor [Nelson, 1998].

The determination of thermal properties is divided into steady-state experiments and the transient experiments. A steady-state experiment is used to evaluate the conductivity properties, and a pair of transient experiments is used to evaluate the thermal diffusivity of the materials used to fabricate the detector and the Seebeck coefficient corresponding to the reference and active junctions.

Preliminary steady-state simulation of the engineering prototype thermal response provided crucial information about the thermal conductivity of the different materials. The temperature gradients through the junction layers are significantly smaller than those through the absorber and thermal impedance layers. As a matter of fact, it would be difficult if not impossible to detect the small temperature difference ($\sim 10^{-8}\,\mathrm{K}$) encountered across the junctions. From Fourier's law, heat conduction in the absence of a temperature gradient across the junctions implies the existence of an "infinite" thermal conductivity. Although no practical material can attain this condition at room temperature, it is closely approximated by the small thermal resistance within the junctions relative to that of the absorber and thermal impedance layers. In conclusion, the thermal conductivity of the thermal impedance layer, k_3, and absorber layer, k_6, are the candidate properties whose values are to be estimated during the steady-state experiments. These candidate property values, or estimated effective thermal conductivity values, include any contact resistance between the layers.

The transient thermal simulation of the engineering prototype operation provides significant information about the system response over time and, therefore, the thermal diffusivity of the different materials used in the fabrication of the detector. Since the transient response of the engineering prototype is governed by the mass and thermal resistance of the system, the absorber and the thermal impedance layers contribute the most to the time response at the "top" of the system, i.e. at the illuminated surface of the absorber layer. In order to quantify the influence of the junctions, the time response was computed at the top of two models: the engineering prototype model with junctions included, and the engineering prototype model without junctions. The time response difference between the two models, shown in Figure 3.8, quantifies the influence of the junctions. At time 0.006 s, the temperature difference between the two models approaches 25 μK. Although this small difference proves to be a temperature measurement challenge, reliable temperature measurements can be attained by increasing the thickness of the junctions without sacrificing the operation of the prototype detector. Notice that 25 μK represents up to 8 percent of the steady-state temperature at the top of the prototype detector and therefore it is recommended not to be neglected. Hence, the thermal diffusivity of the thermal impedance layer, α_3, of the absorber, α_6, and an "effective" combined thermal diffusivity of each junction, α_j, are the candidate properties to be determined by the transient experiments.

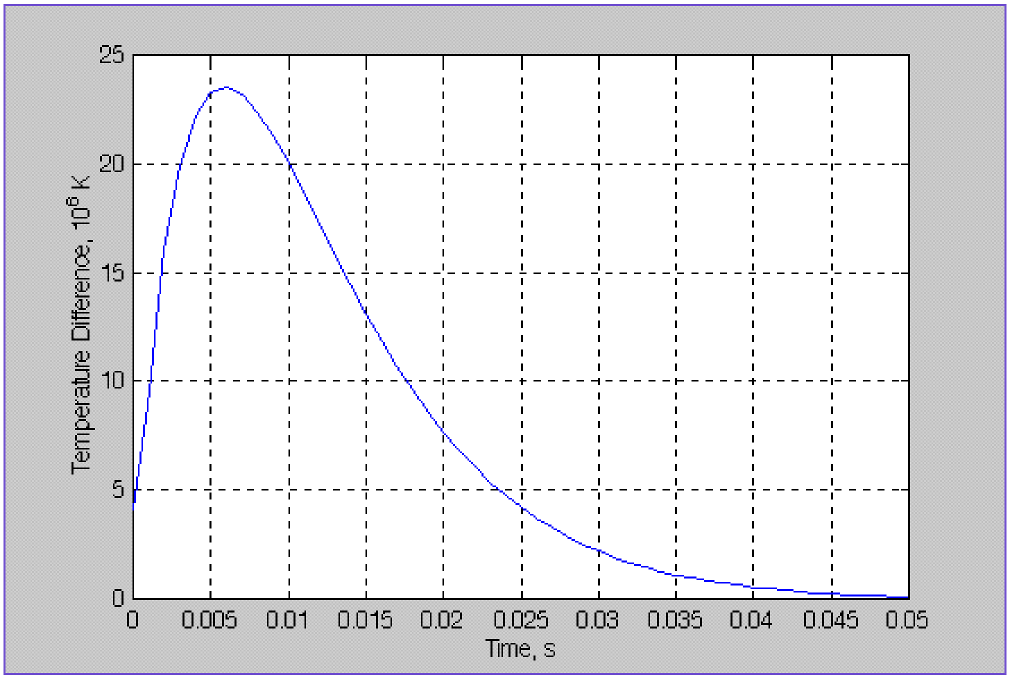
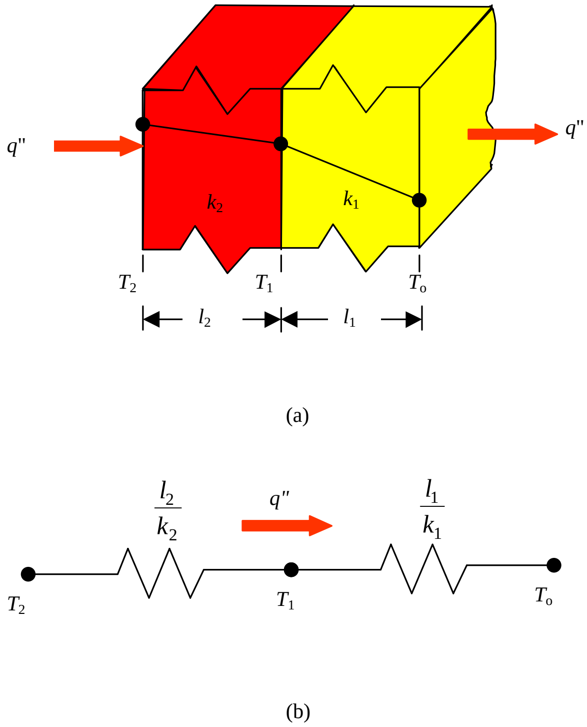


Figure 3.8 Temperature difference between the engineering prototype model without junctions and the same model with junctions, as a function of time.

The "effective" thermal diffusivity of each junction, α_j, assumes a single value for both the zinc-antinomide and platinum layers. By representing the composite medium of the junction as an equivalent two-element thermal circuit in series, as shown in Figure 3.9, the corresponding nominal value of the "effective" thermal conductivity of each junction can be defined as

$$k_j = \frac{l_1 + l_2}{\frac{l_1}{k_1} + \frac{l_2}{k_2}} = \frac{l_4 + l_5}{\frac{l_4}{k_4} + \frac{l_5}{k_5}} \tag{3.7}$$

where $l_1 = l_4$ is the thickness of the platinum layers and $l_2 = l_5$ is the thickness of the zinc-antinomide layers.



(a)

(b)

Figure 3.9 Analogy between conductive heat transfer and electrical current flow [Pitts and Sisson, 1977].

Once k_j has been calculated, the nominal value of α_j for each junction can be determined as

$$\alpha_j = \frac{k_j\left(l_1 + l_2\right)}{\rho_1 c_1 l_1 + \rho_2 c_2 l_2} = \frac{k_j\left(l_4 + l_5\right)}{\rho_4 c_4 l_4 + \rho_5 c_5 l_5}\,, \tag{3.8}$$

where ρ refers to the mass density and c refers to the specific heat. Based on the nominal values presented in Table 1.1, the corresponding values of k_j and α_j are 65.3 W/m·K and 30.88×10^{-6} m/s^2, respectively.

3.3.2 Considerations for Determining the Thermoelectric Properties

The fundamental theorem of thermoelectricity states that the Seebeck effect is the algebraic sum of the thermal Peltier and Thomson effects. In practice, predicted temperatures based on theory and measured voltages do not always provide reliable accuracy. Rather, Doeblin [1990] suggests that a given thermocouple be calibrated over the overall temperature range of the thermocouple application. In this calibration, the Peltier and the Thomson effects are not determined individually but only their overall effect, that is the electromotive force (emf) due to the Seebeck effect. Assuming no irreversible effects such as the Joule effect, the emf generated by a thermocouple is primarily due to be sum of the Peltier emf and Thomson emf. The Peltier emf is assumed to be proportional to the temperature difference between the junctions while the Thomson emf is assumed to be proportional the difference of between the squares of the junction temperatures [Doeblin, 1990]. The general equation expressing the expected emf generated by the engineering prototype detector can be expressed

$$E_{ZnSb-Pt} = S_P\left(T_a - T_r\right) + S_T\left(T_a^2 - T_r^2\right), \tag{3.9}$$

where $E_{ZnSb-Pt}$ is the emf generated by the thermocouple composed of the materials zinc-antinomide and platinum, S_P is the calibration constant related to the Peltier effect, S_T is the calibration constant related to the Thomson effect, T_a is the temperature of the active junction, and T_r is the reference junction temperature. Then given the calibration curve corresponding to Equation 3.9, the Seebeck coefficient can be approximated as

$$\alpha_{ZnSb-Pt}(T) = \lim_{\Delta T \to 0} \frac{\Delta E_{ZnSb-Pt}}{\Delta T} = \frac{dE_{ZnSb-Pt}}{dT}. \tag{3.10}$$

Calibration curves for most thermocouples closely approximate a linear behavior due to the fact that the Peltier effect is normally much greater that the Thomson effect.

3.3.3 Minimization Method Used for Estimation of Thermophysical Properties

Analytical, calculus-based, enumerative, random, and evolutionary methods are some of the possible search and optimization techniques used in parameter estimation problems. For practical reasons, the first four techniques are discarded as possible search tool solutions for the current investigation. First of all, the analytical method requires a differentiable objective function. Although this method might be seen as a possible search tool, the complexity of the differentiation of the objective function is neither attractive nor practical for the current investigation. Second, the calculus-based, enumerative and random methods show lack of efficiency [Goldberg, 1989]. Therefore, the current investigation utilizes the evolutionary method called *genetic algorithms*. These algorithms are known as powerful, robust search tools that consist of a few relatively simple computational steps.

Genetic algorithms are based on a population of solutions. A population of possible solutions is created and allowed to evolve to an optimal state. The word "genetic" is used because this approach simulates some of the biological mechanics involved in natural selection and natural genetics [Goldberg, 1989].

A typical genetic algorithm starts by randomly creating a set (or "population") of possible solutions (or "individuals"). These sets of solutions are structured vectors consisting of the parameters ("or chromosomes") to be optimized. Each possible solution to the problem is tested in an objective function (or "biological environment") to determine its ability (or "fitness") to solve the problem. Individuals with higher fitness are more likely to become "parents". That is, they are "mated" with members of the population based on their fitness, or ability, to solve the problem. Whether biological or computational, at least three genetic operations must occur to influence the evolution of the population. These operations are: (1) reproduction, (2) crossover, and (3) mutation. The reproduction operation is used to mate parents. The crossover operation is used to exchange chromosomes between parents resulting in the formation of a new population, that is, a new generation. Finally, the mutation operation is used to introduce random chromosome changes to the new generation to assure a high level of diversity. After the new generation has been born, the genetic algorithm repeats the biological operations to evolve a sequence of new populations from one "generation" to the next one until an optimal state (or "evolutionary adaptation") has been reached. This optimal state occurs when a solution (or "individual") has satisfied (or "adjusted to") the objective function requirements (or "to its defined biological environment"). In other words, a solution of the parameter estimation problem is reached when it minimizes the objective function to a specified degree of accuracy.

The three most important aspects of using genetic algorithms are the definition and implementation of the objective function, the genetic encoding, and the genetic operators. In this investigation, the objective function is defined as the derivative of the deviation function, and the objective is satisfied when this function becomes zero. The deviation function, or the so-called general sum-of-squares function, can be defined as the sum of the squares of the differences between experimentally determined and mathematically predicted functions, and is quantitatively described as

$$S = \sum_{j=1}^{m} \left(\eta_{E,j} - \eta_{T,j} \right)^{2} . \qquad (3.11)$$

In Equation 3.10 η_T is the predicted value, η_E is the experimental observation, m is the number of observations, and S is the deviation function to be minimized.

The genetic encoding can use any representation for the individual chromosomes in the genetic algorithm. While individuals are often represented using strings of binary code, the current study uses individuals represented as vectors consisting of the parameters, such as k_3 and k_6, to be optimized. Since each of these vectors represents a complete solution to the problem, two MATLAB[1] codes have been developed to implement genetic algorithm routines of the steady-state and transient simulated experiments. Although MATLAB is an

[1] MATLAB (by MathWorks, Inc.) is a numerical computation, data analysis, and visualization software.

excellent tool for numerical computations, it showed lack of efficiency and practicability for executing complex genetic algorithms in a computationally fast manner. The genetic algorithm routine for the steady-state experiments uses a MATLAB function called **gassk36.m** (Appendix B) and considers individuals whose chromosomes consist of values of the variables $k_3$and k_6. The genetic algorithm routine for the transient experiments uses a MATLAB function called **gataj36.m** (Appendix C) and considers individuals whose chromosomes consist of values of the variables α_j, α_3, and α_6. While the genetic encoding is different, the genetic operators are the same for both genetic algorithm codes.

The implementation of the genetic operators involves a two–step interactive process: fitness evaluation and execution of the genetic operators. The genetic algorithm generates at random an initial population comprising individuals, each representing a possible solution of the problem. The fitness evaluation starts by evaluating the solution produced by using the parameter values corresponding to each of the individuals. In the current investigation, an individual is a vector containing the thermal properties to be optimized and the solution is the predicted temperature value. A high fitness is attributed to the individual whose parameter set provides a close fit between predicted values and "measured data". Recall that in this study the experiment is simulated using the numerical model of the engineering prototype, and so "data" are actually computed rather than measured. After the fitness evaluation, genetic operators, a set of simple computational operations, are introduced in order to evolve the initial population and generate successive populations that improve over time. The genetic operators considered in **gassk36.m** and **gataj36.m** are *reproduction, crossover,* and *mutation*. Wright [1996] explains that these operators serve to perform both the diversity and selection requirements in the evolutionary process. While diversity is necessary to explore new areas in the search space, selection is essential to ensure the convergence of the solution.

Reproduction is the process in which individuals are mated according to their fitness. Mating according to fitness means that an individual with higher fitness values has a greater possibility of being chosen as a parent. The reproduction operator implemented in this investigation was based on the “roulette” technique. This technique assumes that the area of the "roulette" wheel assigned to a given individual is proportional to the fitness (refer to Figure 3.10). So, when spinning the roulette wheel, the probability of selecting a mate depends strictly on the fitness.

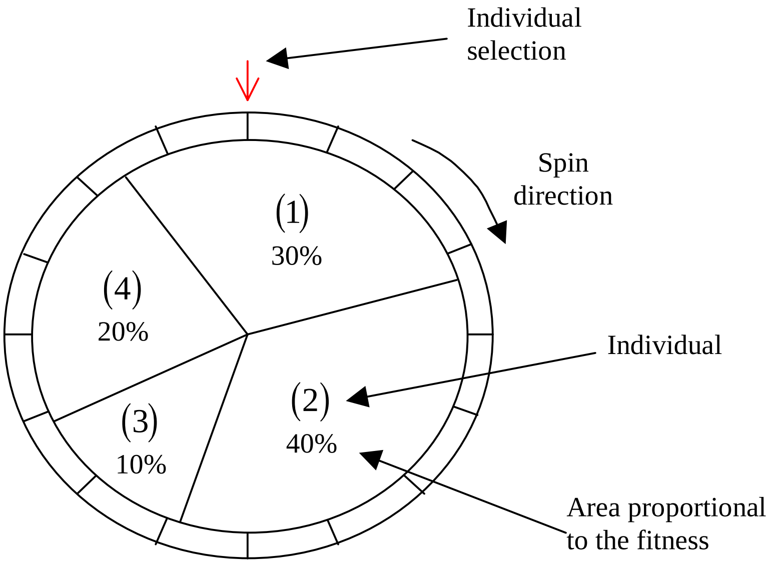


Figure 3.10 Roulette wheel with slots sized according to fitness [Goldberg, 1989].

After reproduction, or selection of parents, the crossover operator is used to determine the selection of offspring chromosomes. The chromosome selection occurs in three different ways: the new chromosome is an exact replica of the chromosome of either Parent One or Parent Two, or the new chromosome is the average value between the chromosome of Parent One and Parent Two.

After the crossover operator is implemented, each chromosome in the new generation is checked for mutation. Although the probability of mutation is normally small, the mutation, or alteration of chromosomes at random, plays an important role in searching the potential parameter values whose location in parameter space could not otherwise be reached by only combining the reproduction and crossover operators.

At this point, a new and improved generation has evolved and the two–step interactive process (fitness evaluation and execution of the genetic operators) is repeated several times until either one of two criteria is met: either the maximum number of generations or the optimal state is reached.

4. Simulation of Experiments

This chapter describes numerical simulations used to estimate the electro-thermophysical properties of the thermal radiation detector. Experimental data are not yet available; therefore, the current study is completely theoretical. The purpose behind the numerical simulations is primarily to provide insight into how the accuracy of the measurement system and the nature of the experiments can influence the estimation of the electro-thermophysical properties.

The chapter starts by presenting the experimental procedure used to simulate the estimation of the electro-thermophysical properties. Then, the results of several simulated experiments are presented and discussed.

4.1 Analysis

Fluctuations of measured data during experimental trials are the result of random errors of the measurement system. A considerable number of accurate measurements are necessary to approximate the exact values within acceptable limits. In theory, an infinite number of measurements must be obtained and averaged when estimating the properties of a system with zero error. Therefore, the need to determine the number of samples required to accomplish the estimation of properties to within an acceptable degree of accuracy is apparent. The number of samples, or observations, depends not only on the nature and procedure of the experiment but also on the uncertainty associated with the measurements. In the following sections the procedure for simulating the experiments and obtaining simulated measurements is explained.

4.1.1 Simulation of the Experimental Procedure

As discussed in Chapter 2, the parameter estimation procedure described in this thesis is accomplished indirectly through a combination of modeling and experimental data. Once again, it is emphasized that the "experimental" part of the procedure is accomplished through simulation in the work reported here. Determination of electro-thermophysical properties is divided into three parts: a set of steady-state experiments that provide essential information about the thermal conductivities, a set of transient experiments that provide information the thermal diffusivities, and a set of transient experiments that provide information about the Seebeck coefficient. In determining the thermal conductivities k_3 and k_6, a sample consisting of temperature measurements at the top of the experimental model, at the hot junction and at the substrate are required during the steady-state experiment, as shown in Figure 4.1. In determining the thermal diffusivities α_j, α_3 and α_6, a sample consisting of a temperature measurement at the substrate and at the top of the experimental model at five time intervals is carried out during a transient experiment. The times considered during the simulated experiment are 0, 0.004, 0.006, and 0.008 s and steady state. The locations of the corresponding temperature measurements are shown in Figure 4.2. Finally, in determining the Seebeck coefficient $\alpha_{ZnSb-Pt}$, a sample consisting of temperature measurements at the substrate and at the top of the absorber layer, the latter at two different times, is carried out during a transient experiment. The times that are considered during the corresponding simulated experiments are after 0.001 s and steady state. Recall that at this point the behavior of the calibration curves is uncertain because no experiment has yet been conducted. In this particular simulation, the behavior of the calibration curve is assumed to follow an ideal case, that is a linear behavior. If actual calibration curves prove to be nonlinear, the number of temperature measurements will need to be increased. The locations of the corresponding temperature measurements are shown in Figure 4.3.

While estimation of the properties requires only a single sample, the accuracy of the estimates depends on the number of samples. Therefore, the dependence of the degree of accuracy of the estimates on the uncertainty of the individual temperature measurements must be determined. In order to better estimate the number of samples needed to acquire a given degree of accuracy, each of the experiments was carried out three times while accumulating a realistic number of samples each time.

Figure 4.4 presents the procedure followed during the experimental simulations.

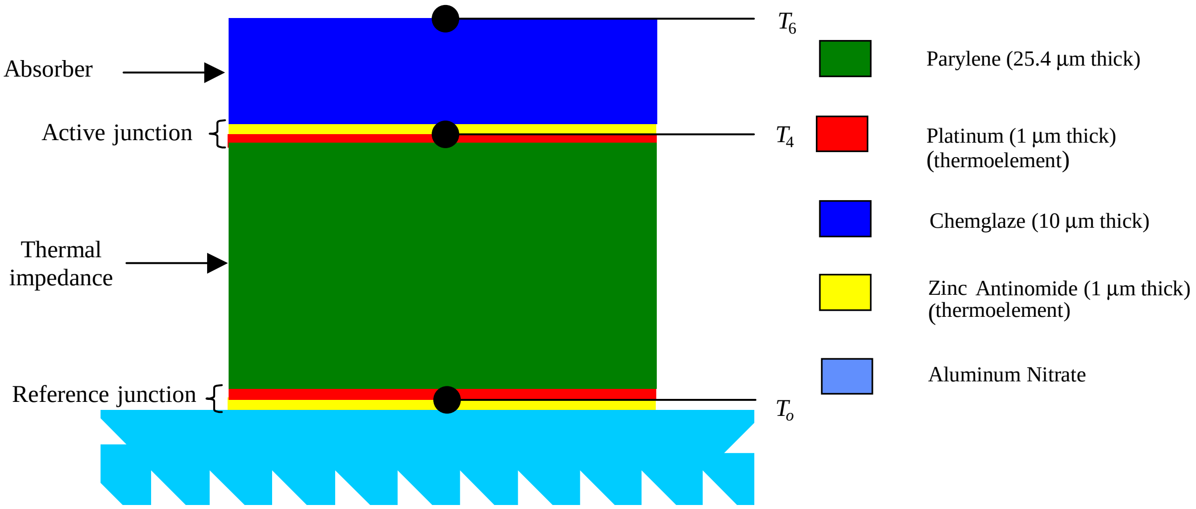


Figure 4.1 Steady-state experiment for determining thermal conductivities.

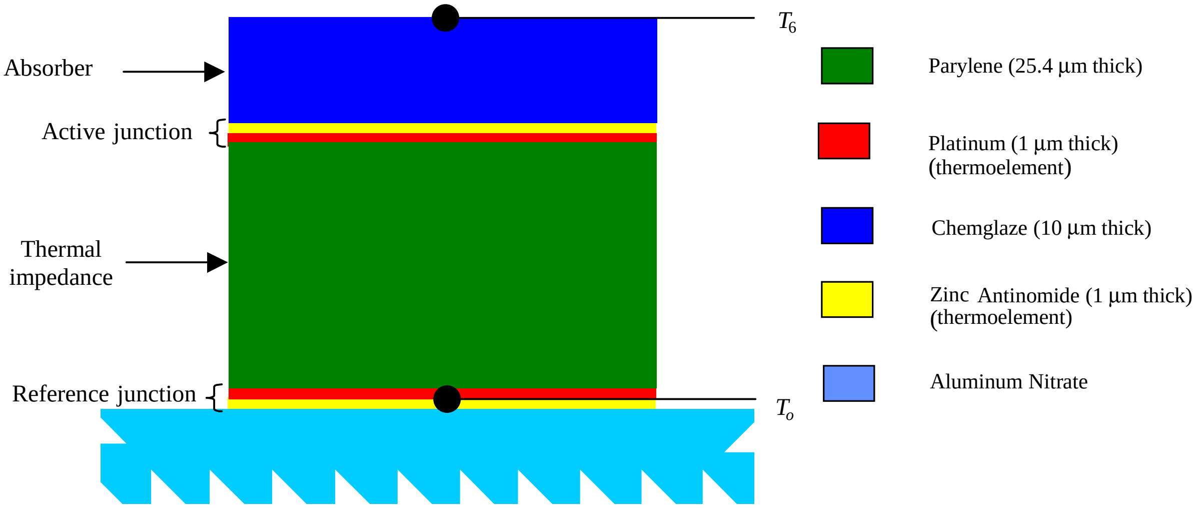


Figure 4.2 Transient experiment for determining thermal diffusivities.

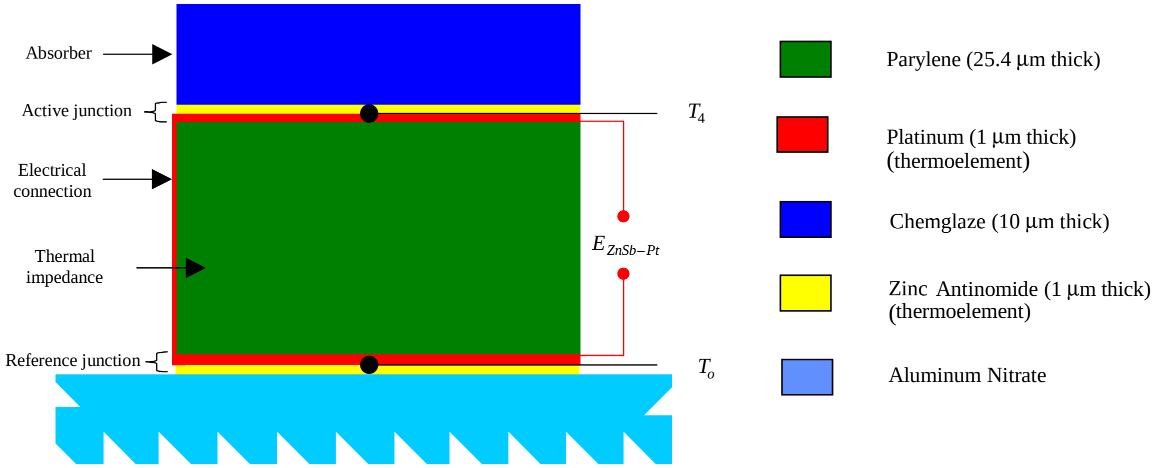


Figure 4.3 Transient experiment for determining the Seebeck coefficient.

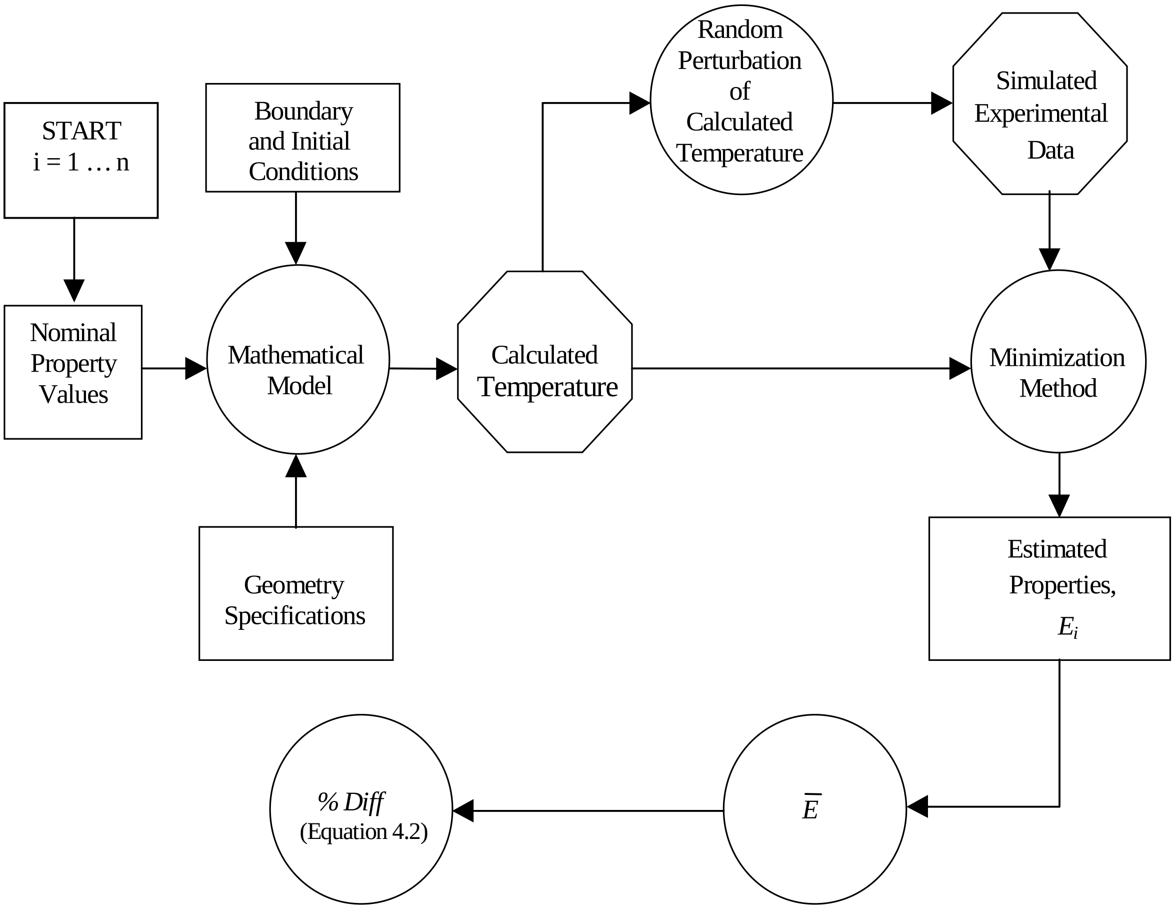


Figure 4.4 Experimental simulation procedure.

In each of the experiments, n samples are obtained. The simulation starts by introducing the nominal values of the property to be estimated, the boundary conditions, the initial conditions and dimensional specifications into the mathematical model of the engineering prototype detector. The nominal property values are given in Table 1.1, the boundary and the initial conditions are presented in Section 3.2.1, and the dimensional specifications are presented in Figure 3.1. As previously discussed, depending on the experiment, temperatures are calculated at different times and locations. These temperatures are randomly perturbed (refer to Section 4.1.2) in order to simulate temperature fluctuations that would be encountered during actual experimental measurements. A minimization method, a genetic algorithm, is used to estimate the physical property values corresponding to the perturbed temperatures. The values of the estimated properties E_i are stored and the average value $\overline{E}$ is updated. The average values of the estimated properties are computed as

$$\overline{E} = \frac{1}{n}\sum_{i=1}^{n} E_i \, , \tag{4.1}$$

where n is the total number of samples considered during the experiments. Finally, after the average value $\overline{E}$ has been calculated, the relative percent difference between the nominal value E_{nom} and the average value $\overline{E}$ of the property values is computed as

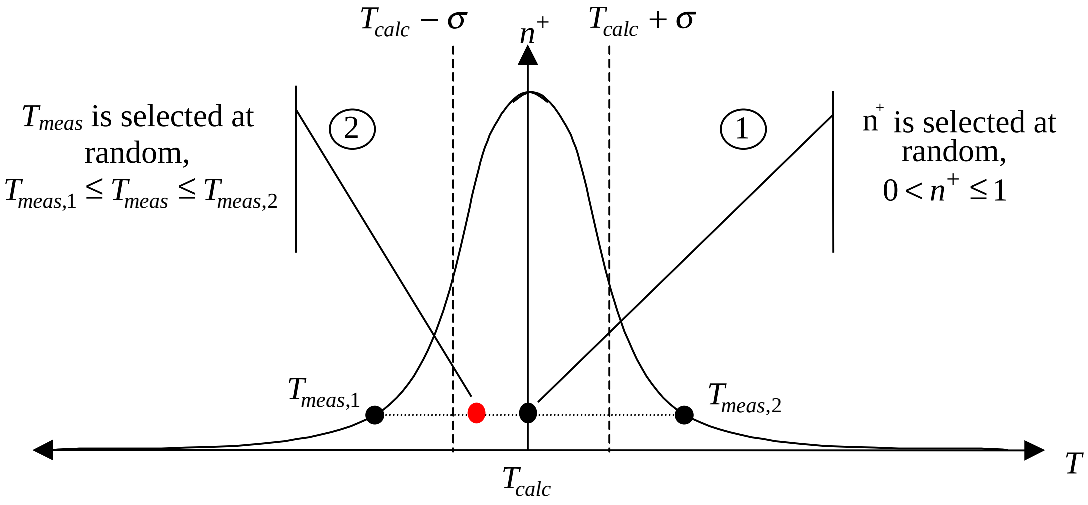
$$\% Diff = \left| \frac{E_{nom} - \overline{E}}{E_{nom}} \right| \times 100 \, . \tag{4.2}$$

The results of the above procedure are presented and discussed in Section 4.2. The objective is to evaluate how accurately a given number of samples n can recover the nominal properties.

The method for obtaining simulated temperatures is presented in the next section.

4.1.2 Simulation of Experimental Data

Sample fluctuations can be characterized by assigning the uncertainty inherent to the measurement system. This is accomplished by normally perturbing the mean value according to the uncertainty associated with the temperature measurements.



$T_{calc} - \sigma$
n^+
$T_{calc} + \sigma$
T_{meas} is selected at random,
$T_{meas,1} \le T_{meas} \le T_{meas,2}$
2
1
n^+ is selected at random,
$0 < n^+ \le 1$
$T_{meas,1}$
$T_{meas,2}$
T_{calc}
T

Figure 4.5 Normal distribution n^+

During the simulations, the sampling population is assumed to follow a normal distribution, shown in Figure 4.5. Given a normal distribution of measured temperatures T_{meas}, the distribution curve can be expressed by the normalized density function

$$n^+ = \frac{n(T_{meas})}{n_{\max}} = e^{-(1/2)\left[(T_{meas}-T_{calc})/\sigma\right]^2}, \quad -\infty < T_{meas} < \infty \tag{4.3}$$

where $n_{\max} = \frac{1}{\sqrt{2\pi\sigma}}$ and σ is the standard deviation. The calculated temperature T_{calc} is the mean value of the simulated temperature because it corresponds to the nominal values of the properties to be estimated during the simulated experiments.

When solving in Equation 4.3 for the random variable T_{meas}, the limits

$$T_{meas,1} = \sigma^2 \left(\frac{T_{calc}}{\sigma^2} + \frac{\sqrt{-2\ln\left(n^+\right)}}{\sigma} \right) \qquad (4.4)$$

and

$$T_{meas,2} = \sigma^2 \left(\frac{T_{calc}}{\sigma^2} - \frac{\sqrt{-2\ln\left(n^+\right)}}{\sigma} \right) \qquad (4.5)$$

are defined. Note that limits $T_{meas,1}$, $T_{meas,2}$ are symmetrical about point T_{calc}.

Given the relative uncertainty $\frac{\sigma}{T_{calc}}$, the perturbed temperature measurements can be determined as

$$T_{meas} = T_{calc} + N_{rand}\left(T_{meas,2} - T_{meas,1}\right) \qquad (4.6)$$

or

$$T_{meas} = T_{calc} + \left(1 - 2\,N_{rand}\right)\sqrt{-2\sigma^2 \ln\left(n^+\right)} \qquad \text{for } 0 < \text{n}^+ \leq 1, \quad (4.7)$$

where n^+ and N_{rand} are uniformly distributed random numbers.

4.2 Results and Discussion

The results of the simulated experimental procedure are presented and discussed in this section. In general, the generic algorithm is demonstrated to be a reliable minimization method. Recall that the genetic algorithm is used to provide the minimal deviation of the theoretical response of the heat conduction model (Section 3.2) from simulated experimental data. As expected, a larger number of samples provides a closer approximation to the nominal values. In this investigation, a value of $\% \, Diff$ of 0.1 percent has been defined as the target value of accuracy for the thermal properties while a value of $\% \, Diff$ of 0.2 percent has been defined as the target value of accuracy for the Seebeck coefficient. Although several measurement uncertainties were considered, the investigation focuses primarily on the results based on an uncertainty of ± 1.0 percent.

Table 4.1 presents a summary of the minimum number of samples required to reach the target values of accuracy in the case of a measurement uncertainty of ± 1.0 percent.

The corresponding results for the estimation of the thermal conductivity and diffusivity values and the Seebeck coefficient are discussed separately in the following sections.

4.2.1 Estimation of the Thermal Conductivity

The graphical results are presented in Figure 4.6 and Figure 4.9. These contain the results of three experiments involving one- and five-percent uncertainty on the simulated temperatures.

Figure 4.6 and Figure 4.7 correspond to the graphical results involving a one-percent uncertainty in the calculated temperature. A % *Diff* of ± 0.1 percent is reached for a sample number of $n =$ 200 while a % *Diff* of ± 0.05 percent is reached for a sample number of $n = 500$. The results of the simulated experiments corresponding to $n = 200$ and an uncertainty of ± 1.0 percent are presented in Table 4.2.

Figure 4.8 and Figure 4.9 correspond to the results involving a five-percent uncertainty in the calculated temperature. A % *Diff* of ± 0.5 percent is reached for a sample number of $n =$ 250 while a sample number of $n = 3000$ is required to obtain a % *Diff* of ± 0.1 percent.

4.2.2 Estimation of the Thermal Diffusivity

The results for the thermal diffusivity estimates are presented between Figure 4.10 and Figure 4.15. These contain the results of three experiments involving ± 0.25 and ± 1.0 percent uncertainty on the calculated temperatures. As the cumulative number of samples is increased, each of these experiments follows a similar behavior.

Figure 4.10 and Figure 4.12 correspond to the results involving 0.25-percent uncertainty on the calculated temperature. A % *Diff* = 0.1 percent is attained for a sample number $n = 500$, while a % *Diff* = 0.05 percent is attained for a sample number $n = 1000$.

Figure 4.13 and Figure 4.15 correspond to the results involving a one-percent uncertainty on the calculated temperature. A % *Diff* = 0.5 percent is attained for a sample number $n = 250$, while a % *Diff* = 0.1 percent is attained for a sample number $n = 2000$.

The results of the simulated experiments corresponding to $n = 2000$ and an uncertainty of ± 1.0 percent are presented in Table 4.3.

4.2.3 Estimation of the Seebeck Coefficient

The results for the estimation of the Seebeck coefficient are presented in Figure 4.16. These contain the results for three experiments involving one-percent uncertainty on the calculated temperatures and voltages.

Figure 4.16 corresponds to the results involving a one-percent uncertainty in the calculated temperature. A $\%\,Diff$ = 0.5 percent is attained for a sample number $n = 200$, while a $\%\,Diff$ = 0.1 percent is attained for a sample number $n = 2000$. The results of the simulated experiments corresponding to $n = 2000$ and an uncertainty of ± 1.0 percent are given Table 4.4

Table 4.1 Recommended sample number *n* (uncertainty in temperature measurements = ± 1.0 percent).

Property	Target % *Diff* [*]	Number of Samples Required, n
k_3, k_6	± 0.1	200
$\alpha_j, \alpha_3, \alpha_6$	± 0.1	3000
$\alpha_{ZnSb-Pt}$	± 0.2	2000

$$^{*}\%Diff = \left|\frac{E_{nom} - \overline{E}}{E_{nom}}\right| \times 100$$

Table 4.2 Relative difference between the nominal value and the sample mean of the estimated values of k_3 and k_6 during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent) , $n = 200$.

	Nominal Values	Estimated Values $n = 200$		
		Exp. 1	***Exp. 2***	***Exp. 3***
k_3 (W/m·K)	0.0840	0.0840 ± 0.0001	0.0840 ± 0.0000	0.0840 ± 0.0001
% Diff[**]	0 %	0.05 %	0.02 %[*]	0.08 %
k_6 (W/m·K)	0.2090	0.2090 ± 0.0001	0.2090 ± 0.0001	0.2090 ± 0.0000
% Diff[**]	0 %	0.04 %	0.06 %	0.01 %[*]

[*] This difference does not appear in the first four significant figures.

$$^{**}\%Diff = \left|\frac{E_{nom} - \overline{E}}{E_{nom}}\right| \times 100$$

Table 4.3 Relative difference between the nominal value and the sample mean of the estimated values of α_j, α_3 and α_6 during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent), $n = 2000$.

	Nominal Values	**Estimated Values** $n = 2000$		
		Exp. 1	***Exp. 2***	***Exp. 3***
α_j (m^2/s)	30.88	30.88 ± 0.02	30.88 ± 0.03	30.88 ± 0.03
% Diff**	0 %	0.05 %	0.09 %	0.08 %
α_3 (m^2/s)	0.0915	0.0915 ± 0.0001	0.0915 ± 0.0001	0.0915 ± 0.0001
% Diff**	0 %	0.05 %	0.09 %	0.05 %
α_6 (m^2/s)	0.2232	0.2232 ± 0.0000	0.2232 ± 0.0000	0.2232 ± 0.0001
% Diff**	0 %	0.01 %*	0.01 %*	0.05 %

* This difference does not appear in the first four significant figures.

$$^{**}\%\,Diff = \left| \frac{E_{nom} - \overline{E}}{E_{nom}} \right| \times 100$$

Table 4.4 Relative difference between the nominal value and the sample mean of the estimated values of $\alpha_{ZnSb-Pt}$ during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent), $n = 2000$.

	Nominal Values	Estimated Values $n = 2000$		
		Exp. 1	***Exp. 2***	***Exp. 3***
$\alpha_{ZnSb-Pt}$ (μV/K)	920.0	920.0 ± 1.7	920.0 ± 1.5	920.0± 0.7
% Diff[*]	0 %	0.18 %	0.16 %	0.08 %

$$^{*}\%Diff = \left|\frac{E_{nom} - \overline{E}}{E_{nom}}\right| \times 100$$

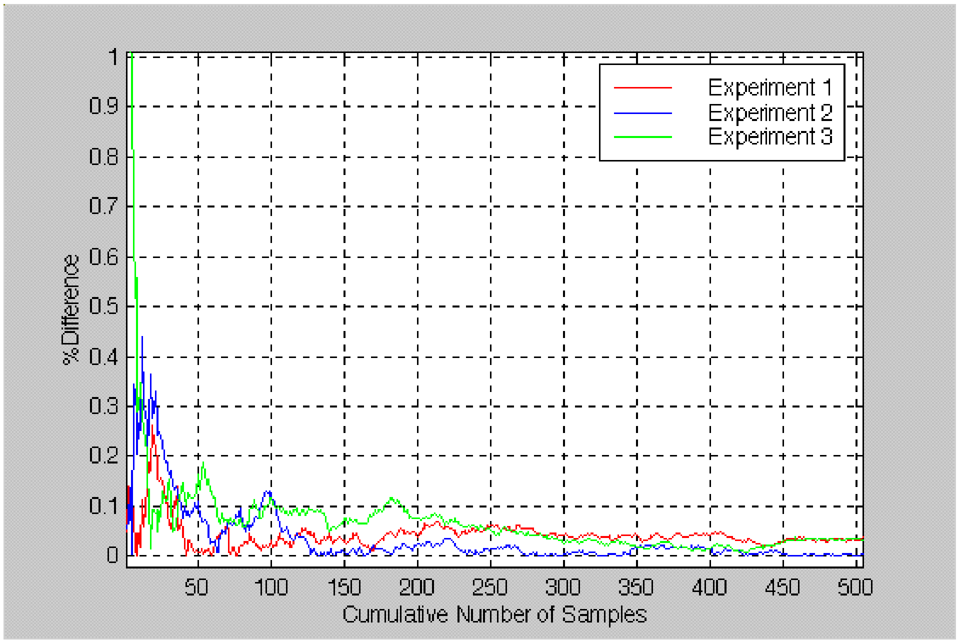


Figure 4.6 Relative difference between the nominal value and the sample mean of the estimated values of k_3 during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent).

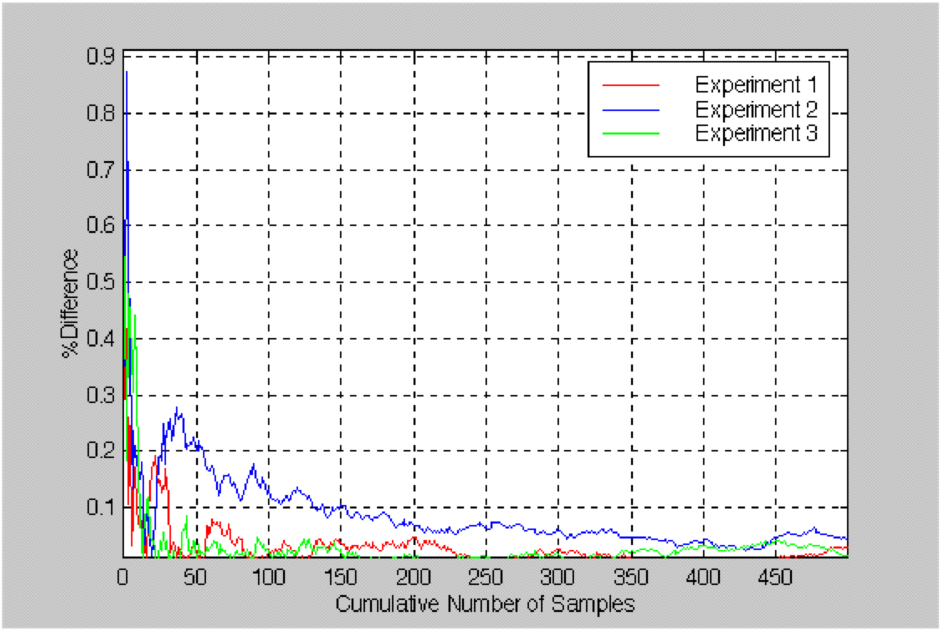


Figure 4.7 Relative difference between the nominal value and the sample mean of the estimated values of k_6 during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent).

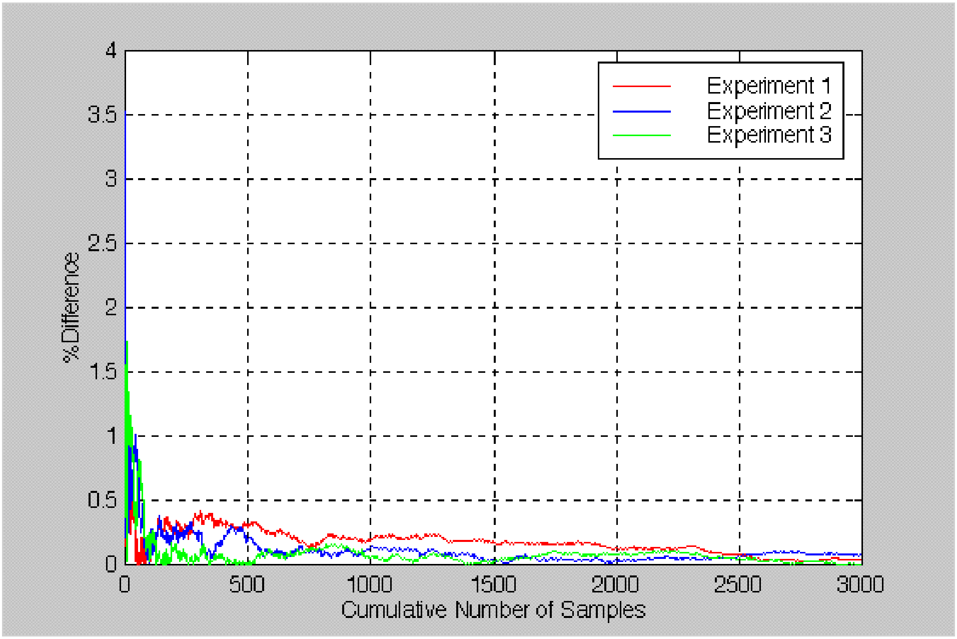


Figure 4.8 Relative difference between the nominal value and the sample mean of the estimated values of k_3 during simulated experiments (uncertainty in temperature measurements = ± 5.0 percent).

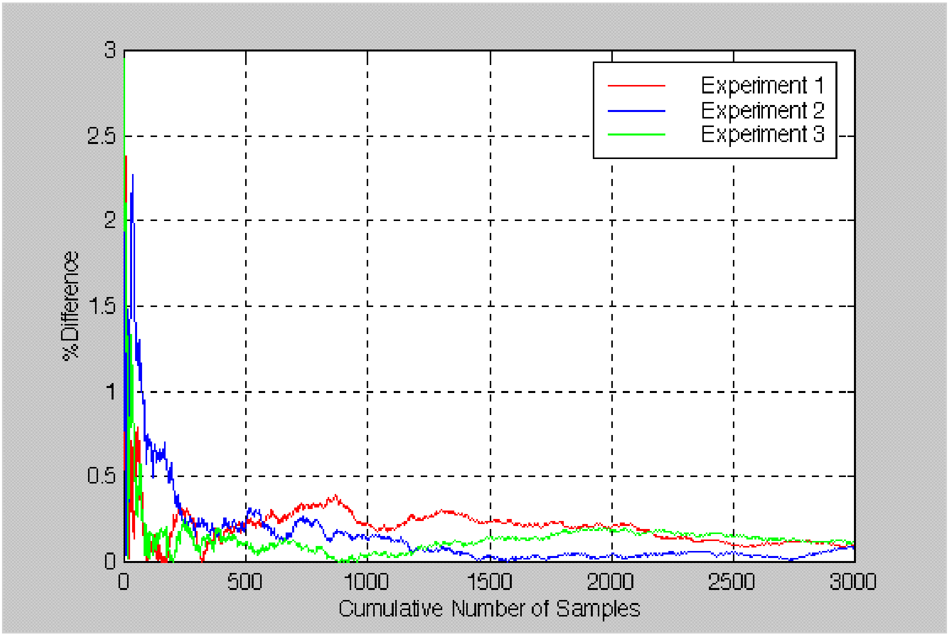


Figure 4.9 Relative difference between the nominal value and the sample mean of the estimated values of k_6 during simulated experiments (uncertainty in temperature measurements = ± 5.0 percent).

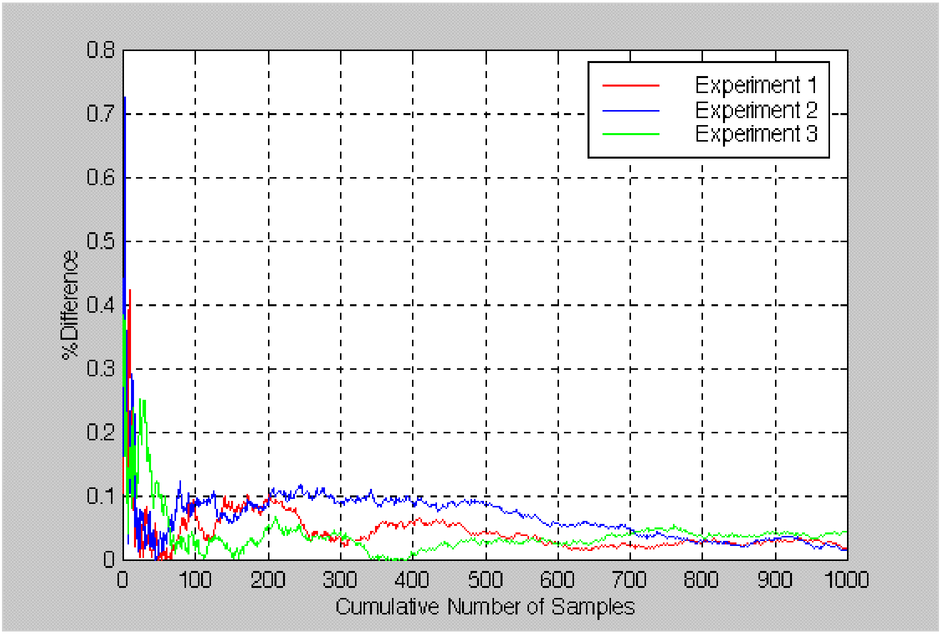


Figure 4.10 Relative difference between the nominal value and the sample mean of the estimated values of α_j during simulated experiments (uncertainty in temperature measurements = ± 0.25 percent).

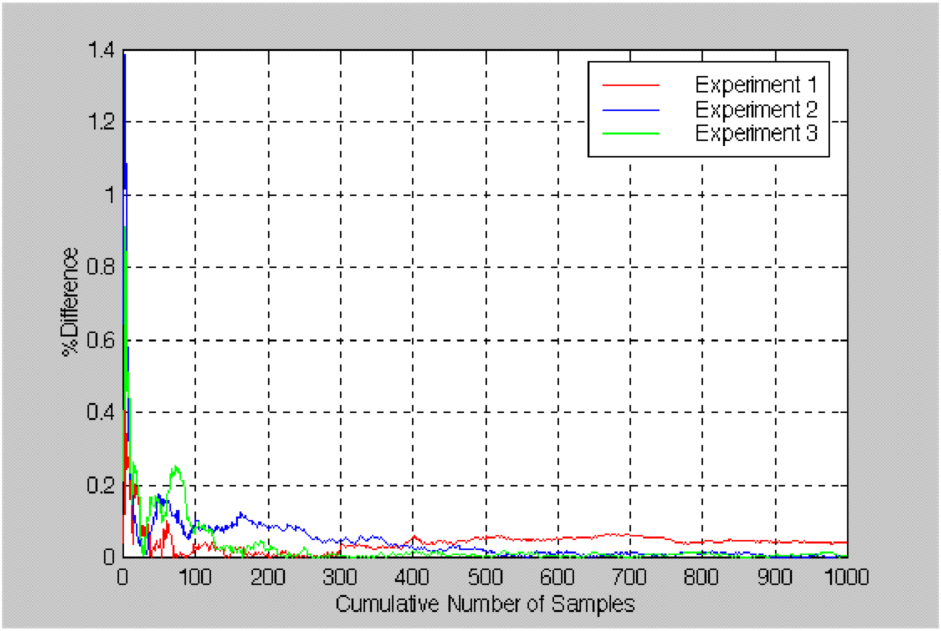


Figure 4.11 Relative difference between the nominal value and the sample mean of the estimated values of α_3 during simulated experiments (uncertainty in temperature measurements = $\pm$0.25 percent).

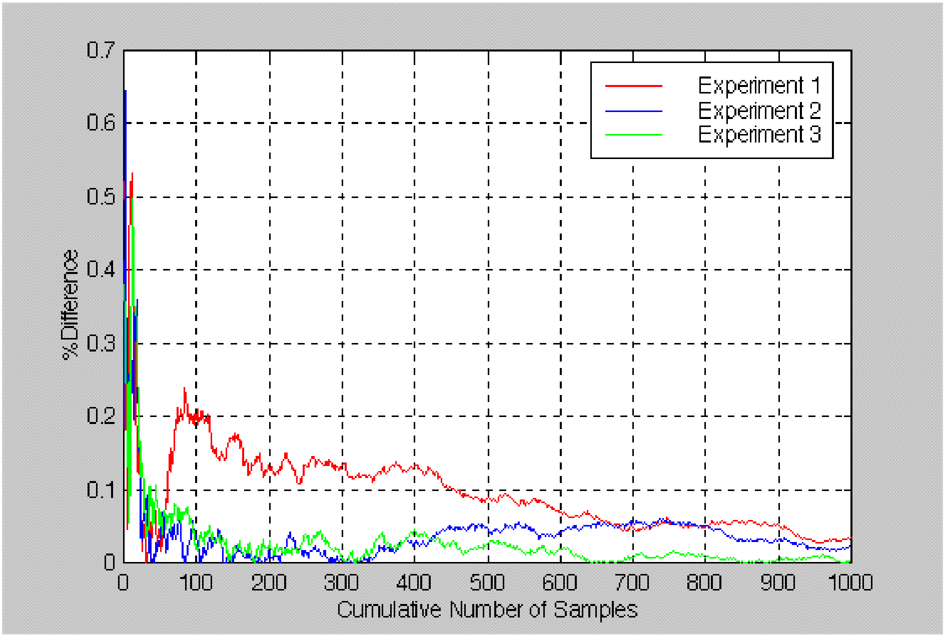


Figure 4.12 Relative difference between the nominal value and the sample mean of the estimated values of α_6 during simulated experiments (uncertainty in temperature measurements = $\pm$0.25 percent).

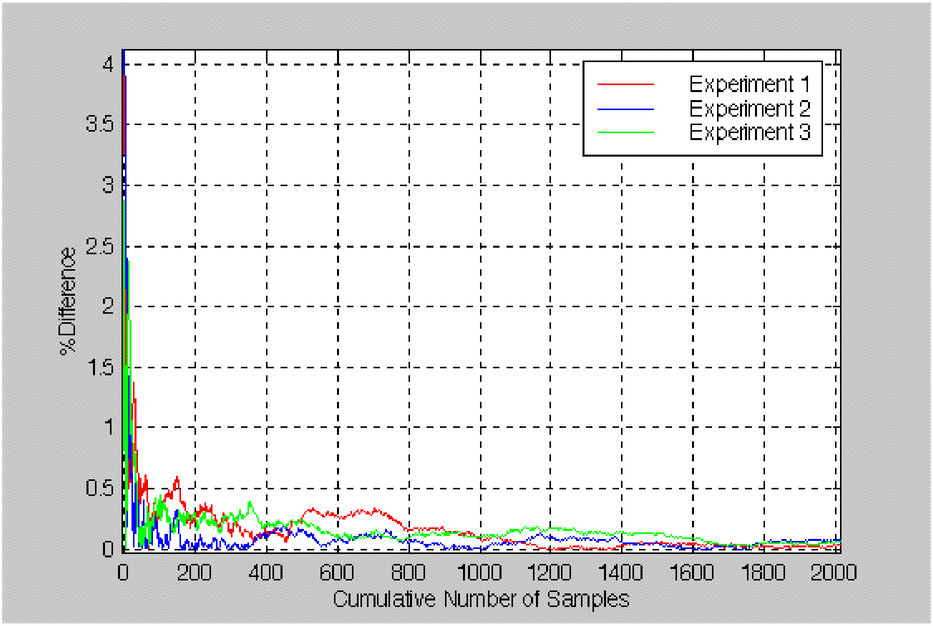


Figure 4.13 Relative difference between the nominal value and the sample mean of the estimated values of α_j during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent).

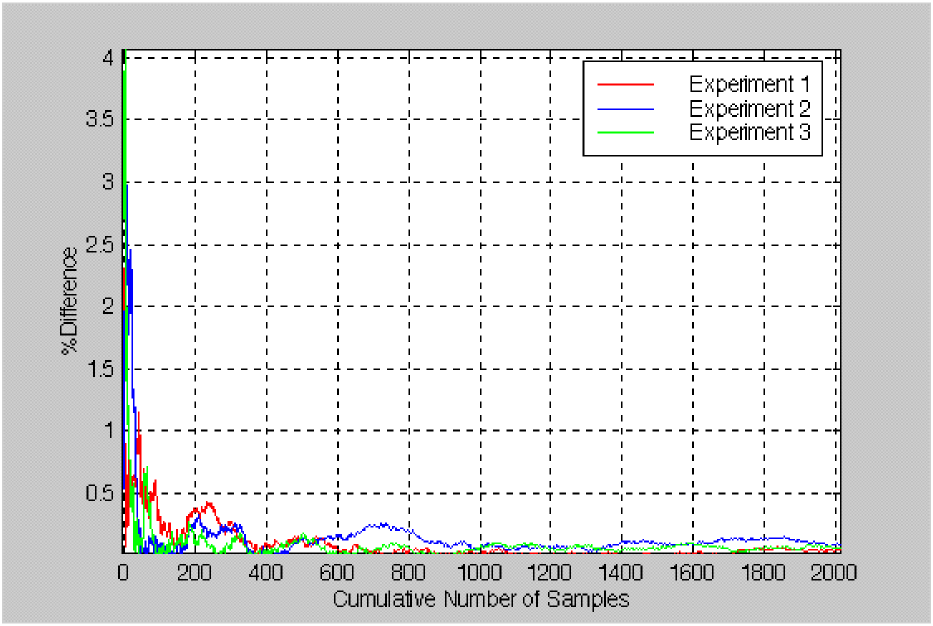


Figure 4.14 Relative difference between the nominal value and the sample mean of the estimated values of α_3 during simulated experiments (uncertainty in temperature measurements = ± 1.0 percent).

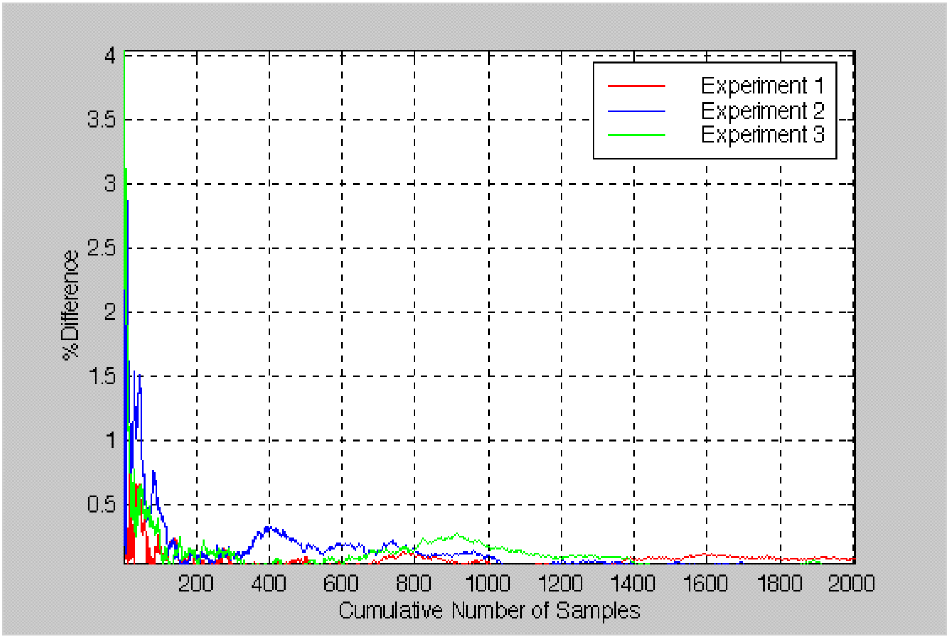


Figure 4.15 Relative difference between the nominal value and the sample mean of the estimated values of α_6 during simulated experiments (uncertainty in temperature measurements = ±1.0 percent).

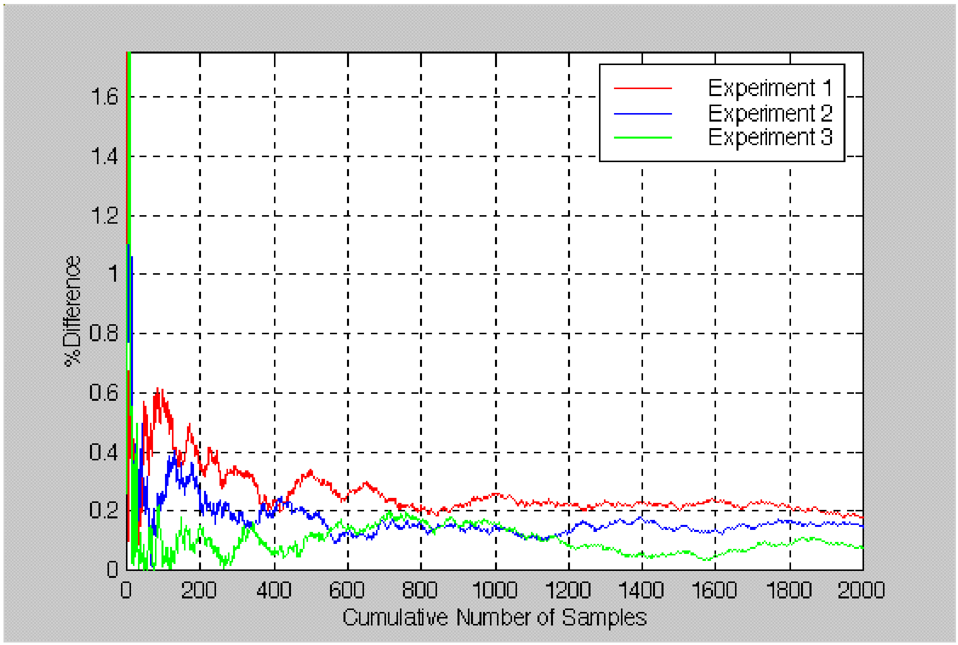


Figure 4.16 Relative difference between the nominal value and the sample mean of the estimated values of $\alpha_{ZnSb-Pt}$ during simulated experiments (uncertainty in temperature and voltage measurements = ±1.0 percent).

5. Conclusions and Recommendations

The primary objectives of this study were to design an experimental procedure to be used for the estimation of electro-thermophysical properties of the different materials used to fabricate a certain thermopile thermal radiation detector. These properties include the thermal conductivity and diffusivity and the Seebeck coefficient. Simulated experiments were conducted using the experimental procedure in order to approximate the number of samples required to achieve an acceptable degree of accuracy.

5.1 Conclusions

Based on the work presented in this thesis, it may be concluded that:

1. An engineering prototype detector was designed to incorporate the dimensional specifications and functional capabilities of the actual thermal radiation detector. The engineering prototype can play the role of the thermal radiation detector while providing the ability to accurately predict its dynamic electrothermal behavior.

2. A genetic algorithm was formulated that proved to be a reliable minimization method for the parameter estimation procedure.
3. The estimation of electro-thermophysical properties can be simplified by estimating the thermal and electrical properties separately.
4. The analytical model corresponding to the engineering prototype detector is in agreement with ALGOR finite element method model. The analytical model can be used as a powerful tool for the development of the thermal radiation detector.
5. The absorber and thermal impedance layers dominate the temperature distribution through the engineering prototype detector. The temperature gradient through the junctions is extremely small, implying an effectively "infinite" thermal conductivity.
6. While the thermal response of the analytical model of the engineering prototype is dominated by the absorber and thermal impedance layers, the combination of the active and passive junctions can contribute up to eight percent of temperature difference at 0.006 s after a step heat flux input. This condition favors consideration of an "effective" combined thermal diffusivity of the two junctions.

5.2 Recommendations

Based on the work presented, it is recommended that:

1. The MATLAB program **gataj36** should be translated into another programming language such as C++ or Fortran. Although MATLAB is an excellent programming language, it carries too much overhead for the intensive computing demanded by the genetic algorithm developed for this investigation. A program in C++ or Fortran will be helpful to obtain the estimation of properties in a less computationally intensive manner.

2. An investigation should be performed using actual experimental data to verify that the behavior of the calibration curve T_{meas} versus $E_{ZnSb-Pt}$ is linear, as assumed in this investigation.

References

Angrist, S. W., 1965, Direct Energy Conversion, Allyn and Bacon Inc., Boston.

Beck, J. V. and K. J. Arnold, 1977, Parameter Estimation in Engineering and Science, John Wiley & Sons, New York.

Goldberg, D. E., 1989, Genetic Algorithms In Search, Optimization, And Machine Learning, Addison-Wesley Pub. Co., Massachusetts.

Krieder, K. G., 1994, Intermetallic Thermocouples, United States Patent, Patent Number 5,356,485.

Mahan, J. R. and L. W. Langley, 1996, The Geo-Synchronous Earth Radiation Budget Instrument: A Thermopile Linear Array Thermal Radiation Detector, Proposal submitted to NASA, Hampton, VA.

Mahan, J. R. and L. W. Langley, 1997, Technology Development of a Sputtered Thermopile Thermal Radiation Detector for Earth Radiation Budget Applications, Proposal submitted to NASA, Hampton, VA.

Mikhailov, M. D. and M. N. Ozisik, 1984, Unified Analysis and Solutions of Heat and Mass Diffusion, John Wiley and Sons Inc., New York.

Muller, J., 1997, http://www.sp.ph.ic.ac.uk/~johannes/GERB.html.

Nelson, E., 1998, Personal Communication, Vatell Corporation, Blacksburg, VA.

Lambropoulos, J. C., S. D. Jacobs, S. J. Burns, L. Shaw-Klein and S.-S. Hwang, 1991, Thermal Conductivity on Thin Films: Measurement and Microstructural Effects, HTD-Vol. 184, Thin Film Heat Transfer: Properties and Processing, ASME, New York, NY.

Oliver, M. B., 1998, http://ssdnt01.bnsc.rl.ac.uk/gerb/web_page.htm

Ozisik, M. N., 1993, Heat Conduction, John Wiley and Sons Inc., New York.

Pitts, D. R., and L. E. Sisson, 1977, Heat Transfer (Schaum's Outline Series), McGraw-Hill, Inc.

Pollock, D. D., 1982a, Physical Properties of Materials for Engineers, Vol. 1, 2nd Ed., CRC Press Inc., Boca Raton, FL.

Pollock, D. D., 1982b, Physical Properties of Materials for Engineers, Vol. 2, 2nd Ed., CRC Press Inc., Boca Raton, FL.

Pollock, D. D., 1982c, Physical Properties of Materials for Engineers, Vol. 3, 2nd Ed., CRC Press Inc., Boca Raton, FL.

Sánchez, M. C., 1998, Optical Analysis of a Linear-Array Thermal Radiation Detector for Geostationary Earth Radiation Budget Applications, M.S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State university, Blacksburg, VA.

Scott, E. P., 1997, Personal Communication, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Tauc, J., 1961, Photo and Thermoelectric Effects in Semiconductors, Pergamon Press Ltd., London.

Weckmann, S., 1997, Dynamic Electrothermal Model of a Sputtered Thermopile Thermal Radiation Detector for Earth Radiation in Budget Applications, M.S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Wright, A. D., 1996, Acoustic Boundary Condition Estimation in a Near-Scale Gas-Turbine Combustor, M.S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Appendix A. MATLAB Program attemp6

This program, *attemp6*, is used to calculate the transient temperature of the experimental model shown in Figure 3.3. The flow chart of *attemp6* is presented in Figure A.1.

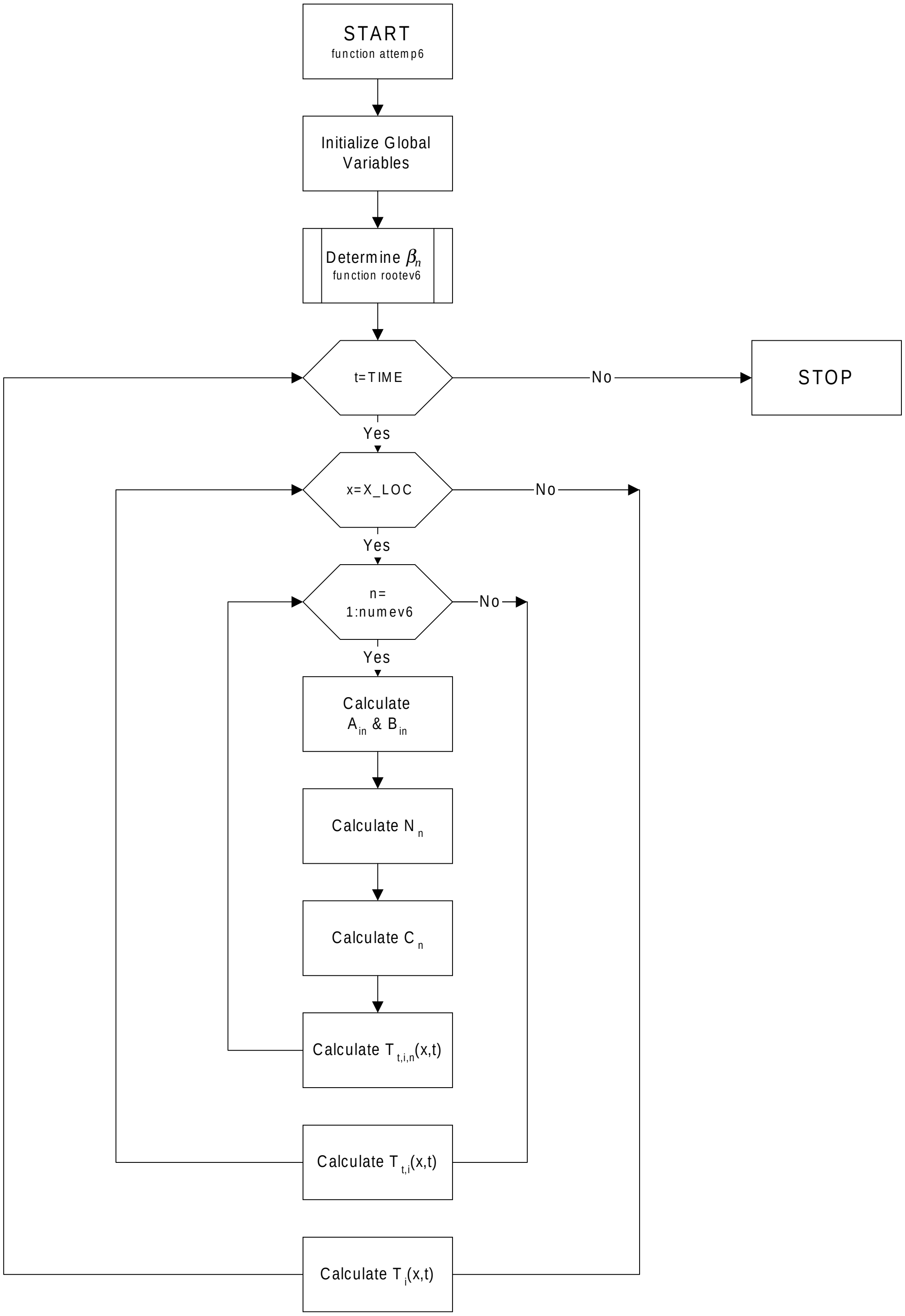


Figure A.1 Flow chart of MATLAB function *attemp6*.

MATLAB file: attemp6sc.m

```
% ATTEMP6SC includes system information used in ATTEMP6.
% ATTEMP6 solves for the analytical 1-dimensional temperature
% distribution of a 6-layer system.
%
clear,clf,clc
tic
%
%SYSTEM INFORMATION
q=1;%W/m^2, Heat flux at the top of the system.
To=0;%K, Constant temperature at the bottom of system.
L=[1,1,25.4,1,1,10]*10^(-6);  % m,  Thickness of layers.
K=[71.6,60,.084,71.6,60,.209]; %W/m*K,  Conductivity of layers.
A=[25,43.6,.0915,25,43.6,.22315]*10^(-6); %m^2/s,  Diffusivity of layers.
TIME=[.001,.0054,.01,100];  %sec,  Specific times to be considered.
X_LOC=[0,1,2,4,6,8,10,12,14,16,18,20,22,24,...
26,27.4,28.4,29.4,31.4,33.4,35.4,37.4,39.4]*10^(-6);m,  Locations x.
numev6=2; % Number of eigenvalues to be considered.
%
%ANALYTICAL SOLUTION
[tempdist,spacedist]=attemp6(X_LOC,TIME,numev6,q,L,K,A)
%
%PLOTTING
plot(spacedist*10^6,tempdist*10^6)
grid
xlabel('Distance x (10^6 m)')
ylabel('Temperature (10^6 K)')
title('Transcient Temperature Distribution of Analytical Model of
Experimental Design')
axis([0 40 0 400])
legend('.001 sec','.0054','.01','100')
toc
```

MATLAB file: attemp6.m

```
function [TEMPDIST,SPACEDIST]=attemp6(X_LOC,TIME,numev6,Q,L,K,A)
%Purpose:
%        ATEMP6 solves for the analytical 1-dimensional temperature
%        distribution of a 6-layer system.
%
%Syntax:
%        [TEMPDIST,SPACEDIST]=atemp6(X_LOC,TIME,numev6,q,L,K,A)
%
%Description:
%         Input:
%         * X_LOC corresponds to the vector of locations to be evaluated.
%         * TIME corresponds to the vector of times to be evaluated.
%         * numev6 corresponds to the number of eigenvalue solutions to be
considered.
```

```
%         * L corresponds to the vector containing the layer thickness
values.
%         * K corresponds to the vector containing the layer conductivity
values.
%         * A corresponds to the vector containing the layer diffusivity
values.
%
%         Output:
%         ATEMP6 returns matrix TEMPDIST and SPACEDIST.
%         * The rows of TEMPDIST(or temperature distribution)correspond
%         to the the rows of SPACEDIST (or x-locations).
%         * The columns of TEMPDIST and SPACEDIST correspond to the
%         different times.
%
%See also   detev6, rootev6.

global q
global k1 k2 k3 k4 k5 k6
global alpha1 alpha2 alpha3 alpha4 alpha5 alpha6
global x1 x2 x3 x4 x5 x6

q=Q;
x1=L(1); x2=x1+L(2); x3=x2+L(3); x4=x3+L(4); x5=x4+L(5); x6=x5+L(6);
k1=K(1); k2=K(2); k3=K(3); k4=K(4); k5=K(5); k6=K(6);
alpha1=A(1); alpha2=A(2); alpha3=A(3); alpha4=A(4); alpha5=A(5);
alpha6=A(6);

ev=rootev6(numev6,L,K,A);

countert=0;
for t=TIME;
countert=countert+1;

counterx=0;
for x=X_LOC;
counterx=counterx+1;
   D(counterx,countert)=x;
      for i=1:numev6
      beta=ev(i);


eta1=beta*x1/sqrt(alpha1); eta2=beta*x2/sqrt(alpha2);
eta3=beta*x3/sqrt(alpha3); eta4=beta*x4/sqrt(alpha4);
eta5=beta*x5/sqrt(alpha5); eta6=beta*x6/sqrt(alpha6);

kappa12=k1/k2*sqrt(alpha2/alpha1); kappa23=k2/k3*sqrt(alpha3/alpha2);
kappa34=k3/k4*sqrt(alpha4/alpha3); kappa45=k4/k5*sqrt(alpha5/alpha4);
kappa56=k5/k6*sqrt(alpha6/alpha5);

A1n=1;B1n=0;
A2n=-1*(-(A1n*sin(eta1)+B1n*cos(eta1))*sin(x1/x2*eta2)-
kappa12*(A1n*cos(eta1)-B1n*sin(eta1))*cos(x1/x2*eta2));
B2n=-1*(kappa12*(A1n*cos(eta1)-B1n*sin(eta1))*sin(x1/x2*eta2)-
(A1n*sin(eta1)+B1n*cos(eta1))*cos(x1/x2*eta2));
```

```
A3n=-1*(-(A2n*sin(eta2)+B2n*cos(eta2))*sin(x2/x3*eta3)-
kappa23*(A2n*cos(eta2)-B2n*sin(eta2))*cos(x2/x3*eta3));
B3n=-1*(kappa23*(A2n*cos(eta2)-B2n*sin(eta2))*sin(x2/x3*eta3)-
(A2n*sin(eta2)+B2n*cos(eta2))*cos(x2/x3*eta3));
A4n=-1*(-(A3n*sin(eta3)+B3n*cos(eta3))*sin(x3/x4*eta4)-
kappa34*(A3n*cos(eta3)-B3n*sin(eta3))*cos(x3/x4*eta4));
B4n=-1*(kappa34*(A3n*cos(eta3)-B3n*sin(eta3))*sin(x3/x4*eta4)-
(A3n*sin(eta3)+B3n*cos(eta3))*cos(x3/x4*eta4));
A5n=-1*(-(A4n*sin(eta4)+B4n*cos(eta4))*sin(x4/x5*eta5)-
kappa45*(A4n*cos(eta4)-B4n*sin(eta4))*cos(x4/x5*eta5));
B5n=-1*(kappa45*(A4n*cos(eta4)-B4n*sin(eta4))*sin(x4/x5*eta5)-
(A4n*sin(eta4)+B4n*cos(eta4))*cos(x4/x5*eta5));
A6n=-1*(-(A5n*sin(eta5)+B5n*cos(eta5))*sin(x5/x6*eta6)-
kappa56*(A5n*cos(eta5)-B5n*sin(eta5))*cos(x5/x6*eta6));
B6n=-1*(kappa56*(A5n*cos(eta5)-B5n*sin(eta5))*sin(x5/x6*eta6)-
(A5n*sin(eta5)+B5n*cos(eta5))*cos(x5/x6*eta6));

Nn=k1/alpha1*quad8('phinsq',0,x1,[],[],beta,alpha1,A1n,B1n)+...
         k2/alpha2*quad8('phinsq',x1,x2,[],[],beta,alpha2,A2n,B2n)+...
         k3/alpha3*quad8('phinsq',x2,x3,[],[],beta,alpha3,A3n,B3n)+...
         k4/alpha4*quad8('phinsq',x3,x4,[],[],beta,alpha4,A4n,B4n)+...
         k5/alpha5*quad8('phinsq',x4,x5,[],[],beta,alpha5,A5n,B5n)+...
         k6/alpha6*quad8('phinsq',x5,x6,[],[],beta,alpha6,A6n,B6n);

int1=k1/alpha1*quad8('phinf1',0,x1,[],[],beta,alpha1,A1n,B1n);
int2=k2/alpha2*quad8('phinf2',x1,x2,[],[],beta,alpha2,A2n,B2n);
int3=k3/alpha3*quad8('phinf3',x2,x3,[],[],beta,alpha3,A3n,B3n);
int4=k4/alpha4*quad8('phinf4',x3,x4,[],[],beta,alpha4,A4n,B4n);
int5=k5/alpha5*quad8('phinf5',x4,x5,[],[],beta,alpha5,A5n,B5n);
int6=k6/alpha6*quad8('phinf6',x5,x6,[],[],beta,alpha6,A6n,B6n);

Tp1(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha1,A1n,B1n)*(int1+int2+int3+int4+int5+int6);
Tp2(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha2,A2n,B2n)*(int1+int2+int3+int4+int5+int6);
Tp3(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha3,A3n,B3n)*(int1+int2+int3+int4+int5+int6);
Tp4(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha4,A4n,B4n)*(int1+int2+int3+int4+int5+int6);
Tp5(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha5,A5n,B5n)*(int1+int2+int3+int4+int5+int6);
Tp6(i)=1/Nn*exp(-
beta^2*t)*phin(x,beta,alpha6,A6n,B6n)*(int1+int2+int3+int4+int5+int6);
      end

   if x>=0 & x<x1
   Ts=q*x/k1;T(counterx,countert)=Ts-sum(Tp1);
   elseif x>=x1 & x<x2
   Ts=q*(x/k2+x1*(1/k1-1/k2));T(counterx,countert)=Ts-sum(Tp2);
   elseif x>=x2 & x<x3
   Ts=q*(x/k3+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));T(counterx,countert)=Ts-
sum(Tp3);
   elseif x>=x3 & x<x4
   Ts=q*(x/k4+x3*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
   T(counterx,countert)=Ts-sum(Tp4);
   elseif x>=x4 & x<x5
   Ts=q*(x/k5+x4*(1/k4-1/k5)+x3*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-
1/k2));
   T(counterx,countert)=Ts-sum(Tp5);
   elseif x>=x5 & x<=x6
   Ts=q*(x/k6+x5*(1/k5-1/k6)+x4*(1/k4-1/k5)+x3*(1/k3-1/k4)+x2*(1/k2-
1/k3)+x1*(1/k1-1/k2));
   T(counterx,countert)=Ts-sum(Tp6);
   end
   end
end

TEMPDIST=T;
SPACEDIST=D;
```

MATLAB file: asstemp6sc.m

```
% ASSTEMP6SC includes system information used in ATTEMP6.
% ATTEMP6 solves for the analytical steady-state 1-dimensional temperature
% distribution of a 6-layer system.

%SYSTEM INFORMATION
q=1;;%W/m^2, Heat flux at the top of the system.
To=0;%K, Constant temperature at bottom of system.
L=[1 1 25.4 1 1 10]*10^-6;%m, Thickness of layers.
K=[71.6 60 .084 71.6 60 .209];%W/m*K Conductivity of layers.

i=0;
for x=0:.1:39.4
i=i+1;
X_LOC(i)=x*10^-6;
end

%ANALYTICAL SOLUTION
[Tan]=asstemp6(X_LOC,q,L,K);

%PLOTTING
plot(X_LOC*10^6,Tan*10^6)
xlabel('Distance x, (10^6 m)')
ylabel('Temperature, (10^6 K)')
%title('Steady State Temperature Distribution')
grid on
```

MATLAB file: asstemp6.m

```
function [Tan]=asstemp6(X_LOC,q,L,K)
%ATEMP6    Solves for the analytical steady state
%          1-dimensional temperature
%          distribution of a 6-layer system.
%
%FUNCTION:    [Tan]=atemp6(X_LOC,q,L,K)
%
```

```
%          *X_LOC :vector of locations to be evaluated.
%          *L :vector of layer thickness values.
%          *K :vector of layer conductivity values.


x1=L(1);x2=x1+L(2);
x3=x2+L(3);x4=x3+L(4);
x5=x4+L(5);x6=x5+L(6);
k1=K(1);k2=K(2);
k3=K(3);k4=K(4);
k5=K(5);k6=K(6);

counterx=0;
for x=X_LOC;
   counterx=counterx+1;
   if x>=0 & x<x1
   Tan(counterx)=q*x/k1;
   elseif x>=x1 & x<x2
   Tan(counterx)=q*(x/k2+x1*(1/k1-1/k2));
   elseif x>=x2 & x<x3
   Tan(counterx)=q*(x/k3+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
   elseif x>=x3 & x<x4
   Tan(counterx)=q*(x/k4+x3*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
   elseif x>=x4 & x<x5
   Tan(counterx)=q*(x/k5+x4*(1/k4-1/k5)+x3*(1/k3-1/k4)+x2*(1/k2-
1/k3)+x1*(1/k1-1/k2));
   elseif x>=x5
   Tan(counterx)=q*(x/k6+x5*(1/k5-1/k6)+x4*(1/k4-1/k5)+x3*(1/k3-
1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
   end
end

%end atemp6
```

MATLAB file: rootev6.m

```
function [ROOTEV]=rootev6(numev6,L,K,A)

%Purpose:
%          ROOTEV6 solves for the roots of the eigenvalue
%          transcendental equation matrix needed to solve for the
%          temperature distribution of a 6 layer system.
%
%Syntax:
%         [ROOTEV]=rootev6(numev6,L,K,A)
%
%Description:
%         Input:
%         * numev6 corresponds to the number of eigenvalue root solutions
to be considered.
%         * L corresponds to the vector containing the layer thickness
values.
%         * K corresponds to the vector containing the layer conductivity
values.
```

```
%        * A corresponds to the vector containing the layer diffusivity
values.
%
%        Output:
%        * ROOTEV is the vector of root solutions.
%
%Remarks:
%        ROOTEV6 requires function DETEV6
%
% See also   detev6, atemp6


%Scanning eigenvalue function
   counter=0;
   for beta=0:10:500
      counter=counter+1;
      betascan(counter)=beta;
      ev6scan(counter)=detev6(beta,L,K,A);
   end

%Finding the zeros of the eigenvalue (detev6)function.
signflag=sign(ev6scan);
counterev6=0;
for i=2:counter
   if signflag(i)==signflag(i-1)
   dummy1=0;
   else
      counterev6=counterev6+1;
      if counterev6>numev6
         break
      end
   ROOTEV(counterev6)=fzero('detev6',betascan(i),[],[],L,K,A);
   end
end

%Warning Post
if counterev6<numev6
    counterev6
    'WARNING in rootev6.m: numev6 was not satisfied'
    'Solution: increase max value of beta (for loop) during the scanning'
end

%end rootev6
```

MATLAB file: detev6.m

```
function d=detev6(beta,L,K,A)

%Purpose:
%       DETEV6 determines the determinant of the eigenvalue
%       transcendental equation matrix.  The matrix
%       was specifically derived for calculating the
%       temperature distribution of a 6 layer system.
%
```

```
%Syntax:
%       d=detev6(beta,L,K,A)
%
%Description:
%        Input:
%        * beta corresponds to 'eigenvalue' numerical value
%        * L corresponds to the vector containing the layer thickness
values.
%        * K corresponds to the vector containing the layer conductivity
values.
%        * A corresponds to the vector containing the layer diffusivity
values.
%
%        Output:
%        DETEV6 returns the determinant value d
%
%See also   rootev6,atemp6

%Renaming Properties values
x1=L(1); x2=x1+L(2); x3=x2+L(3); x4=x3+L(4);    x5=x4+L(5); x6=x5+L(6);

k1=K(1);k2=K(2); k3=K(3); k4=K(4); k5=K(5); k6=K(6);

alpha1=A(1); alpha2=A(2); alpha3=A(3); alpha4=A(4); alpha5=A(5);
alpha6=A(6);

eta1=beta*x1/sqrt(alpha1); eta2=beta*x2/sqrt(alpha2);
eta3=beta*x3/sqrt(alpha3); eta4=beta*x4/sqrt(alpha4);
eta5=beta*x5/sqrt(alpha5); eta6=beta*x6/sqrt(alpha6);

kappa12=k1/k2*sqrt(alpha2/alpha1); kappa23=k2/k3*sqrt(alpha3/alpha2);
kappa34=k3/k4*sqrt(alpha4/alpha3); kappa45=k4/k5*sqrt(alpha5/alpha4);
kappa56=k5/k6*sqrt(alpha6/alpha5);

%Defining the Eigenvalue Transcendental Function (6 layer system)
D=[sin(eta1),-sin(x1/x2*eta2),-cos(x1/x2*eta2),0,0,0,0,0,0,0,0;...
kappa12*cos(eta1),-cos(x1/x2*eta2),sin(x1/x2*eta2),0,0,0,0,0,0,0,0;...
0,sin(eta2),cos(eta2),-sin(x2/x3*eta3),-cos(x2/x3*eta3),0,0,0,0,0,0;...
0,kappa23*cos(eta2),-kappa23*sin(eta2),-
cos(x2/x3*eta3),sin(x2/x3*eta3),0,0,0,0,0,0;...
0,0,0,sin(eta3),cos(eta3),-sin(x3/x4*eta4),-cos(x3/x4*eta4),0,0,0,0;...
0,0,0,kappa34*cos(eta3),-kappa34*sin(eta3),-
cos(x3/x4*eta4),sin(x3/x4*eta4),0,0,0,0;...
0,0,0,0,0,sin(eta4),cos(eta4),-sin(x4/x5*eta5),-cos(x4/x5*eta5),0,0;...
0,0,0,0,0,kappa45*cos(eta4),-kappa45*sin(eta4),-
cos(x4/x5*eta5),sin(x4/x5*eta5),0,0;...
0,0,0,0,0,0,0,sin(eta5),cos(eta5),-sin(x5/x6*eta6),-cos(x5/x6*eta6);...
0,0,0,0,0,0,0,kappa56*cos(eta5),-kappa56*sin(eta5),-
cos(x5/x6*eta6),sin(x5/x6*eta6);...
0,0,0,0,0,0,0,0,0,cos(eta6),-sin(eta6)];

%Calculating the determinant
d=det(D);

%end detev6
```

MATLAB file: phin.m

```
function y=phin(x,beta,alpha,An,Bn)
%PHIN  is a subfunction of function ATTEMP6
%See also attemp6
y=(An*sin(beta/sqrt(alpha)*x)+Bn*cos(beta/sqrt(alpha)*x));
```

MATLAB file: phinsq.m

```
function y=phinsq(x,beta,alpha,An,Bn)
%PHINSQ  is a subfunction ATTEMP6
%See also attemp6
y=(An*sin(beta/sqrt(alpha)*x)+Bn*cos(beta/sqrt(alpha)*x)).^2;
```

MATLAB file: phinf1.m

```
function [y]=phinf1(x,beta,alpha,An,Bn)
%PHINF1  is a subfunction ATTEMP6
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*x.*q/k1;
```

MATLAB file: phin2.m

```
function y=phinf2(x,beta,alpha,An,Bn)
%PHINF2  is a subfunction of ATTEMP6
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*q.*(x/k2+x1
.*(1/k1-1/k2));
```

MATLAB file: phin3.m

```
function y=phinf3(x,beta,alpha,An,Bn)
%PHINF3  is a subfunction of ATTEMP6
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*q.*(x/k3+x2
*(1/k2-1/k3)+x1*(1/k1-1/k2));
```

MATLAB file: phin4.m

```
function y=phinf4(x,beta,alpha,An,Bn)
%PHINF4  is a subfunction ATTEMP6
```

```
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*q.*(x/k4+x3
*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
```

MATLAB file: phin5.m

```
function y=phinf5(x,beta,alpha,An,Bn)
%PHINF5  is a subfunction ATTEMP6
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*q.*(x/k5+x4
*(1/k4-1/k5)+x3*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
```

MATLAB file: phin6.m

```
function y=phinf6(x,beta,alpha,An,Bn)
%PHINF6 is a subfunction ATTEMP6
%See also attemp6
global  q k1 k2 k3 k4 k5 k6 x1 x2 x3 x4 x5 x6
y=(An.*sin(beta/sqrt(alpha).*x)+Bn.*cos(beta/sqrt(alpha).*x)).*q.*(x/k6+x5
*(1/k5-1/k6)+x4*(1/k4-1/k5)+x3*(1/k3-1/k4)+x2*(1/k2-1/k3)+x1*(1/k1-1/k2));
```

Appendix B. MATLAB Program gassk36

This program, *gassk36*, is the genetic algorithm used to determine the values of k_3 and k_6 for the corresponding measured temperatures. The flow chart of *gassk36* is presented in Figure B.1.

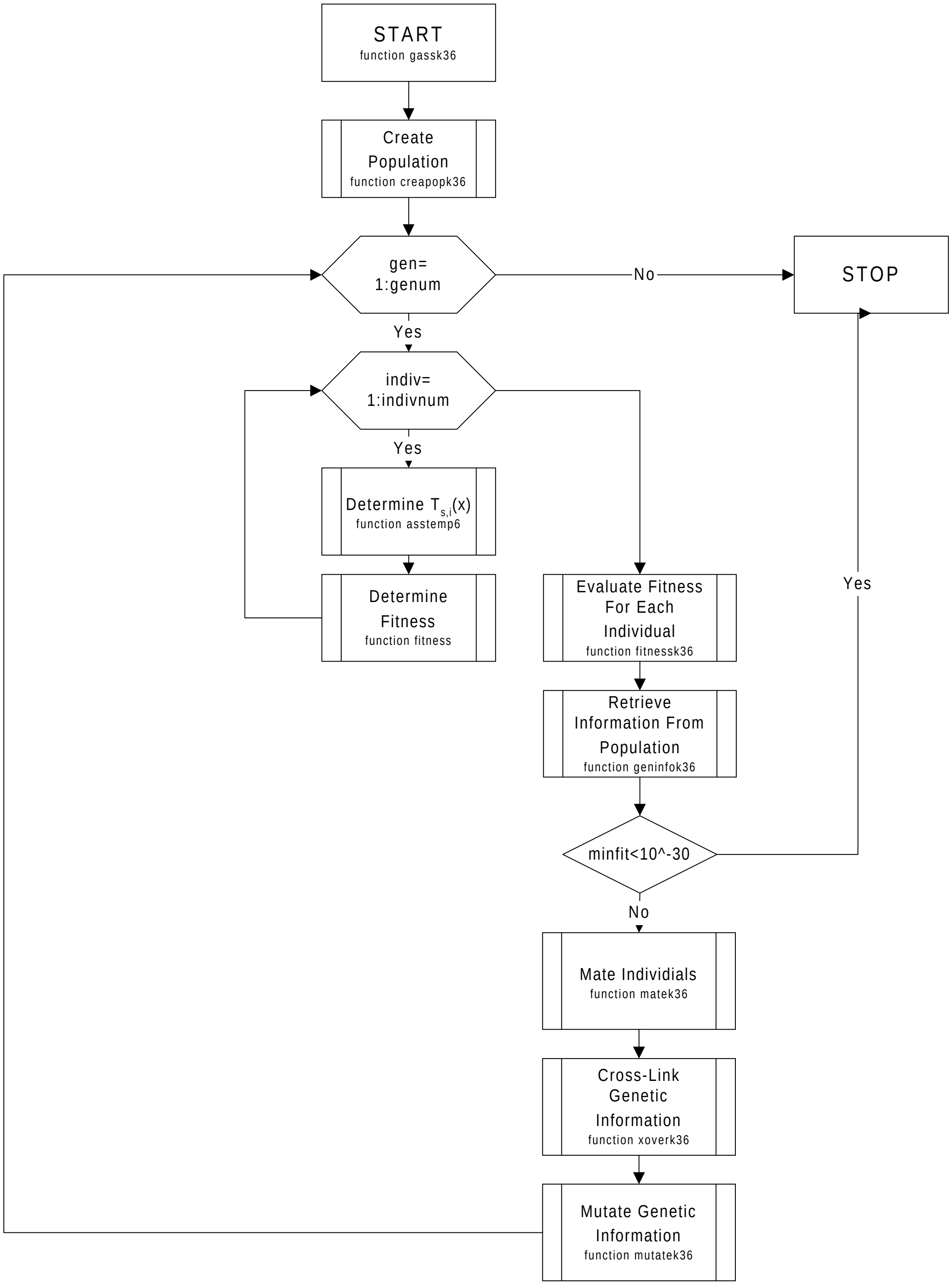


Figure B.1 Flow chart of MATLAB function *gassk36*.

MATLAB file: gassk36sc.m

```
clear
tic
%Assumed To=0
X_LOC=[25.4,35.4].*10^(-6);
q=1;
L=[25.4,10].*10^(-6);
Kguess=[.089,.21]
genum=300;
popnum=20;
uncert=.01;
pmutate=.07;
[Tfem]=asstemp2(X_LOC,q,L,Kguess);
Texper=Tfem;
Kguess=[.0840,.2090]
[BESTK3,BESTK6,gener]=gassk36(genum,popnum,pmutate,Texper,Kguess,X_LOC,q,L
)
toc
```

MATLAB file: gassk36.m

```
function
[BESTK1,BESTK2,gen]=gassk36(genum,popnum,pmutate,Texper,Kguess,X_LOC,q,L)
%Purpose:
%        GASSK36 is the genetic algorithm used to estimate parameters k3
and k6.
%Syntax:
%
[BESTK3,BESTK6,gen]=gassk36(genum,popnum,pmutate,Texper,Kguess,X_LOC,q,L)
%Description:
%         Input:
%         * genum is total number of generations to be evaluated.
%         * popnum is the number of individuals of each population.
%         * pmutate is the probability of mutation.
%         * Texper is a vector containing the experimental temperatures.
%         * Kguess is a vector containing a guess value for k3 and k6.
%         * X_LOC is a vector containing the locations where Texper were
measured.
%         * q is the magnitude of heat flux.
%         * L is a vector containing the thickness values of layer 3 and
6.
%         Output:
%         * BESTK3 and BESTK6 are the estimation values for k3 & k6.
%         * gen is the number of generations needed to evaluate BESTK3 &
BESTK6.
%Remarks:
%         * Convergence is set at a default value of 10^-10.
%         * 10^-6 is the temperature range. 10^-10 is the accuracy.

%Checking number of unknowns==number of measurements.
if length(Kguess)~=length(X_LOC)
'lenght of Kguess ~= length of X_LOC'
```

```
   break
end

%Creating initial population.
[K]=creapopk36(Kguess,popnum);

%Generation loop.
for gen=1:genum

   %Calculating |Texper-Tan| for each individual in population.
   for pop=1:popnum
      [Tan]=asstemp2(X_LOC,q,L,K(pop,:));
      [FIT]=fitness(Tan,Texper);
      FITtrack{pop}=FIT;
      zerotestk1(pop)=FIT(1,1);
      zerotestk2(pop)=FIT(1,2);
   end

   %Checking for possible upcoming invalid division by zero.
   if sum(zerotestk1)==0 |sum(zerotestk2)==0
      gen=gen-1;
      break
   end

   %Calculating fitness.
   [FIT]=fitnessk36(FITtrack,gen);

   %Retrieving information for each population.
   [BESTK1(gen),BESTK2(gen),STATS]=geninfok36(FIT,K);
   minfit(gen)=STATS(1,1); %meanfit(gen)=STATS(1,2);

   %Checking for solution convergence or maximum number of generations.
   if minfit(gen)<10^(-30)
      'A solution has converged. In generation',gen,'Congratulations.'
      break
   elseif gen==genum
      'The maximum number of generations has been reached.'
      break
   end

   %Mating.
   [MATE2,GLOBFIT]=matek36(FIT);

   %Cross-linking.
   [K]=xoverk36(MATE2,FIT,GLOBFIT,Texper,X_LOC,q,L,K);

   %Mutating.
   [K]=mutatek36(K,pmutate);

end

%end gassk36
```

MATLAB file: pertk36 .m

```
function [Texper]=pertk36(Tfem,uncert,notrand)
%Purpose:
%         PERTK36 perturbates the mean value Texp(experimental
temperature)
%         at a given uncertainty(standard deviation/mean value)
%Syntax:
%        [Texper]=pert(Tfem,uncert,notrand)
%Description:
%        Input:
%        * Tfem is vector with FEM temperatures
%        * uncert is the uncertainty to be considered
%        * notrand especifies probability n.  Scaled from 0 to 1 of nmax
%        Output:
%        * Texp is the perturbed Tfem at a given uncertainty and
probability.
%Remarks:
%        * If notrand==0, the a uniformly distributed random number is
chosen.

%Calculating the standard deviation.
stdev=uncert*Tfem;

%Calculating nmax.
nmax(1)=1/sqrt(2*pi)/stdev(1);
nmax(2)=1/sqrt(2*pi)/stdev(2);

%Calculating n.
if notrand>0
   n=nmax*notrand;
elseif notrand==0
   n(1)=nmax(1)*rand;
   n(2)=nmax(2)*rand;
end

%Checking that n is not equal to 0.
if n(1)==0
  n(1)=eps;
elseif n(2)==0
  n(2)=eps;
end

%Calculating Texper.
   Texperf(1)=stdev(1).^2*(1/stdev(1).^2*Tfem(1)+sqrt(-
2*log(n(1)*stdev(1)*sqrt(2*pi)))/stdev(1));
   Texperf(2)=stdev(2).^2*(1/stdev(2).^2*Tfem(2)+sqrt(-
2*log(n(2)*stdev(2)*sqrt(2*pi)))/stdev(2));
   Texperg(1)=stdev(1).^2*(1/stdev(1).^2*Tfem(1)-sqrt(-
2*log(n(1)*stdev(1)*sqrt(2*pi)))/stdev(1));
   Texperg(2)=stdev(2).^2*(1/stdev(2).^2*Tfem(2)-sqrt(-
2*log(n(2)*stdev(2)*sqrt(2*pi)))/stdev(2));

   Texper(1)=Texperf(1)+rand*(Texperg(1)-Texperf(1));
   Texper(2)=Texperf(2)+rand*(Texperg(2)-Texperf(2));
   %end pertk36
```

MATLAB file: creapopk36.m

```
function [K]=creapopk36(Kguess,popnum)
%Purpose:
%        CREAPOPK36 creates a population of parameters k3 & k6.
%Syntax:
%        [K]=creapopk36(Kguess,popnum)
%Description:
%        Input:
%        * Kguess is a vector containing a guess value for k3 and k6.
%        * popnum is the number of populations to be evaluated.
%        Output:
%        * K is a matrix.  Each row represents a combination of k3 and k6.
%Remarks:
%        * A safefactor is used in this function to force band limits.
%          Change this local variable as necessary.
%          Consider to increase pmutate and genum.
%          Default value is .2.


safefactor=.2; %Forcing band limits.
row=popnum;col=length(Kguess);

%Creating random numbers(from 0 to 1).
Kr=rand(row,col)*safefactor;

%Creating random negative values.
K1=ones(row,col);
for i=1:row
   for j=1:col
      if rand>.5
         K1(i,j)=-1;
      end
   end
end

%Creating matrix of mean values and created random deviations.
for i=1:popnum
   Kw(i,:)=K1(i,:).*Kr(i,:).*Kguess(1,:);
   Ku(i,:)=Kguess(1,:);
end

%Creating population by adding mean value and random deviation from mean
value.
K=Ku+Kw;

%end creapopk36
```

MATLAB file: fitness.m

```
function [FIT]=fitness(Tan,Texper)
```

```
%Purpose:
%        FITNESS calculates the temperature difference |Texper - Tan|
%        at the selected location.
%Syntax:
%        [FIT]=fitness(Tan,Texper)
%Description:
%        Input:
%        * Tan is a vector containing the calculated analytical
temperature.
%        * Texper is the a vector containing the experimental temperature.
%        Output:
%        * FIT is a vector containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.
%
%See also pertk36, asstemp2

%Checking vector length for Texper and Tan.
if size(Texper)~=size(Tan)
   'Texp size is not the same as Tan'
   break
end

%Calculating |Texper-Tan|
FIT=abs(Texper-Tan);

%end fitness
```

MATLAB file: fitnessk36.m

```
function [fit]=fitnessk36(FIT,gen)
%Purpose:
%        FITNESSK36 calculates the fitness of each individual in the
population.
%Syntax:
%        [FIT]=fitness(Tan,Texper)
%Description:
%        Input:
%        * Tan is a vector containing the calculated analytical
temperature.
%        * Texper is the a vector containing the experimental temperature.
%        Output:
%        * FIT is a vector containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.
%
%See also pertk36, asstemp2


popnum=length(FIT);

sump1=0;
sump2=0;
```

```
%Weighting equally all chromosomes.
for i=1:popnum
   sump1=(FIT{i}(1,1))+sump1;
   sump2=(FIT{i}(1,2))+sump2;
end

for i=1:popnum
   col1(i)=FIT{i}(1,1)/sump1;
   col2(i)=FIT{i}(1,2)/sump2;
end

%Adding weighted chromosomes for each individual.
sump=(col1+col2);

%Calculating fitness.
fit=sump/2;

%end fit
```

MATLAB file: geninfok36.m

```
function [BESTK1,BESTK2,STATS]=geninfok36(FIT,K)
%Purpose:
%        GENINFO  keeps track of the fittest individual of population.
%Syntax:
%       [BESTK3,BESTK6,MINFIT,STATS]=geninfo(FIT,K)
%Description:
%       Input:
%       * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of chromosomes k3 &
k6.
%       * K is a matrix.  Each row represents a combination of k3 and k6.
%       Output:
%       * STATS=[ minfit meanfit maxfit stdevfit].
%       * BESTK3 & BESTK6 are corresponding chromosomes k3 and k6 of
fittest individual.
%
%See also fitnessk36, creapopk36, mutatek36

%Statistical values.
[minfit,bestind]=min(FIT);
meanfit=mean(FIT);
maxfit=max(FIT);
stdevfit=std(FIT);
STATS=[minfit meanfit maxfit stdevfit];

%Declaring chromosomes of best individual.
BESTK1=K(bestind,1);
BESTK2=K(bestind,2);
```

```
%end geninfok36
```

MATLAB file: matk36.m

```
function [MATE2,GLOBFIT]=matek36(FIT)
%Purpose:
%        MATEK36 mates individuals among population by using a roulette
technique.
%Synntax:
%        [MATE2,GLOBFIT]=matek36(FIT)
%Description:
%        Input:
%        * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of k3 & k6.
%        Output:
%        * MATE2 is a vector containg the row number of mate2.
%        * GLOBFIT is a matrix containing the fitness for each
corresponding
%          combination of k3 & k6 based on FIT.
%
%See also fitnessk36

popnum=size(FIT,2);

%Assigning higher mating probability for small values of FIT.
invfit=FIT.^(-1);
GLOBFIT=invfit/sum(invfit);

%Assigning mate2 to mate1 by using the roulette technique.
for mate1=1:popnum
   rolwheel=rand; %Spinning the roulette./Random hit.
   flag=1;
   selMATE2=0;

   %Finding corresponding mate2 for given mate1
   while flag==1
      selMATE2=selMATE2+1;
      rolocat=sum(GLOBFIT(1:selMATE2));

      if rolocat>=rolwheel
         MATE2(mate1)=selMATE2;
         flag=0;
      end

   end

end

%end mate
```

MATLAB file: xoverk36.m

```
function [Kout]=xoverk36(MATE2,FIT,GLOBFIT,Texper,X_LOC,q,L,K)
%Purpose:
%       XOVERK36 crosslinks information between parents in order to create
%       offsprings.  After mating, the genetic information passed to the
%       offsprings can be done in the following manners: genetic info
%       from parent1, parent2 or the mean value from parent1 & parent2.
%Syntax:
%       function [Kout]=xoverk36(MATE2,FIT,GLOBFIT,Texper,X_LOC,q,L,K)
%Description:
%       Input:
%       * MATE2 is a vector containing the row number of mate2.
%       * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of k3 & k6.
%       * GLOBFIT is a matrix containing the fitness for each
corresponding
%         combination of k3 & k6 based on FIT.
%       * Texper is the a vector containing the experimental temperature.
%       * X_LOC is a vector containing the locations where Texper were
measured.
%       * q is the magnitude of heat flux.
%       * L is a vector containing the thickness values of layer 3 and 6.
%       * K is a matrix containing the population. Each row is an
individual made of k3 & k6.
%       Output:
%       * Kout is a matrix containing the cross-linked population.
%
%See also matek36,fitnessk36

popnum=length(FIT);

for mate1=1:popnum

   %Creating a mean value between parent 1 and parent 2.
   Kout(mate1,:)=(K(mate1,:).*GLOBFIT(mate1)...
   +K(MATE2(mate1),:).*GLOBFIT(MATE2(mate1)))./...
   (GLOBFIT(mate1)+GLOBFIT(MATE2(mate1)));

   %Calculating |Texper-Tan|
   [Tout]=asstemp2(X_LOC,q,L,Kout(mate1,:));
   [FITout]=fitness(Tout,Texper);

   %Cross-linking genetic information directly from parent 1,
   %parent 2, or the mean value between parent 1 & parent 2.
   if FITout<=FIT(mate1)& FITout<=FIT(MATE2(mate1))
      dummy=1;
   elseif FIT(mate1)<FITout & FIT(mate1)<FIT(MATE2(mate1)) & rand<.95
      Kout(mate1,:)=K(mate1,:);
   elseif FIT(MATE2(mate1))<FITout & FIT(MATE2(mate1))<FIT(mate1) &
rand<.95
      Kout(mate1,:)=K(MATE2(mate1),:);
   end

end

%end xover
```

MATLAB file: mutatek36.m

```
function [K]=mutatek36(K,pmutate)
%Purpose:
%       MUTATEK36 mutates one or both of the chromosomes of an individual.
%Syntax:
%       [Kout]=mutatek36(Kin,pmutate)
%Description:
%       Input:
%       * Kin is a matrix containing the population.
%         Each row is an individual made of chromosomes k3 & k6.
%       * pmutate is the probability of mutation.
%       Output:
%       * Kout is a matrix containing the mutated Kin
%
%See also xoverk36

popnum=size(K,1);
laynum=size(K,2);
safefactor=.5; %Defining maximum magnitude of mutation.

%Checking on mutation./Executing mutation.
for i=1:popnum

   for j=1:laynum
      random=rand;
      randsign=rand;

      if  random<=pmutate & randsign<=.5
         K(i,j)=K(i,j)*(1+randsign*safefactor);
      elseif random<=pmutate & randsign>.5
         K(i,j)=K(i,j)*(1-randsign*safefactor);
      end

   end

end

%end mutate
```

Appendix C. MATLAB Program gataj36

This program, *gataj36*, is the genetic algorithm used to determine the values of α_j, α_3 and α_6 for the corresponding measured temperatures. The flow chart of *gataj36* is presented in Figure C.1.

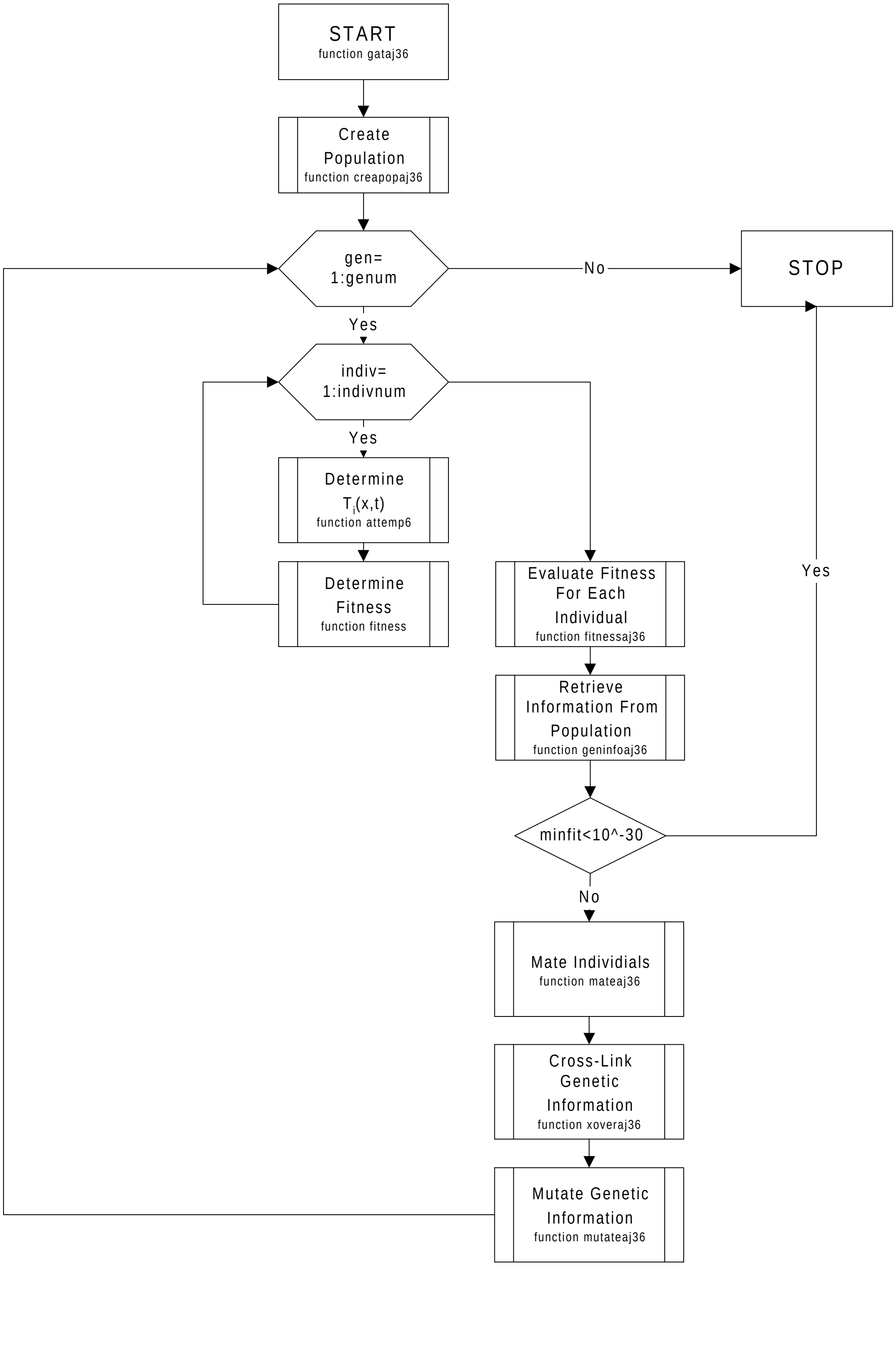


Figure C.1 Flow chart of MATLAB function *gataj36.*

MATLAB file: gatsc.m

```
clear
tic

%Assumed To=0

%(FEM) EXPERIMETAL SYSTEM INFORMATION
X_LOC=[39.4].*10^(-6);
TIME=[.003,.005,.007];
numev6=2;
q=1;
L=[1,1,25.4,1,1,10].*10^(-6);
K=[71.6,60,.084,71.6,60,.209];
A=[30.8778,30.8778,.0915,30.8778,30.8778,.2232]*10^(-6);
[Tfem]=attemp6(X_LOC,TIME,numev6,q,L,K,A);

%GENETIC ALGORITHM INFORMATION
genum=5;
popnum=20;
pmutate=.07;

%RETRIEVAL OF DIFFUSIVITY THERMAL PROPERTIES
%FROM EXPERIMETAL DATA BY USING GENETIC ALGORITHM
uncert=.01;%Uncertainty of temperature measurements
Texper=pertaj36(Tfem,uncert,0);
%Texper=Tfem;
Apar=.0915;
Achem=.2232;
Ajunct=30.8778;
Aguess=[Ajunct,Apar,Achem]*10^(-6);
[BESTAj,BESTA3,BESTA6,gener]=gataj36(genum,popnum,pmutate,Texper,Aguess,X_
LOC,TIME,numev6,q,L,K)
toc
```

MATLAB file: gataj36.m

```
function
[BESTAj,BESTA3,BESTA6,gen]=gataj36(genum,popnum,pmutate,Texper,A3guess,X_L
OC,TIME,numev6,q,L,K)
%Purpose:
%        GATAJ36 is the genetic algorithm used to estimate parameters
aj,a3 and a6.
%Syntax:
%
[BESTAj,BESTA3,BESTA6,gener]=gataj36(genum,popnum,pmutate,Texper,Aguess,X_
LOC,TIME,numev6,q,L,K)
%Description:
%         Input:
%         * genum is total number of generations to be evaluated.
%         * popnum is the number of individuals of each population.
%         * pmutate is the probability of mutation.
%         * Texper is a vector containing the experimental temperatures.
%         * Aguess is a vector containing a guess value for aj, a3 and k6.
%         * X_LOC is a vector containing the locations where Texper were
measured.
%         * q is the magnitude of heat flux.
%         * L is a vector containing the thickness values of all 6 layers.
%         Output:
%         * BESTAj, BESTA3 and BESTA6 are the estimated values for aj, a3
& a6.
%         * gen is the number of generations needed to evaluate BESTAj,
BESTA3 & BESTA6.


%Creating initial population.
[A3]=creapopaj36(A3guess,popnum);

%Generation loop.
for gen=1:genum
   gen
   [A]=three2six(A3);
   %Calculating |Texper-Tan| for each individual in population.
   for pop=1:popnum
      [Tan]=attemp6(X_LOC,TIME,numev6,q,L,K,A(pop,:));
      [FIT]=fitness(Tan,Texper);
      FITtrack{pop}=FIT;
      zerotest1(pop)=FIT(1,1);
      zerotest2(pop)=FIT(1,2);
      zerotest3(pop)=FIT(1,3);
   end

   %Checking for possible upcoming invalid division by zero.
   if sum(zerotest1)==0 | sum(zerotest2)==0 | sum(zerotest3)==0
      break
   end

   %Calculating fitness.
   [FIT]=fitnessaj36(FITtrack,gen);

   %Retrieving information from population.
   [BESTAj(gen),BESTA3(gen),BESTA6(gen),STATS]=geninfoaj36(FIT,A3);
   minfit(gen)=STATS(1,1); %meanfit(gen)=STATS(1,2);

   %Checking for solution based on either convergence or maximum number of
generations.
   if minfit(gen)<10^(-30)
      'A solution has converged. In generation',gen,'Congratulations.'
      break
   elseif gen==genum
      'The maximum number of generations has been reached.'
      break
   end

   %Mating.
   [MATE2,GLOBFIT]=mateaj36(FIT);

   %Cross-linking.
   [A3]=xoveraj36(MATE2,FIT,GLOBFIT,Texper,X_LOC,TIME,numev6,q,L,K,A3);

   %Mutating.
   [A3]=mutateaj36(A3,pmutate);
end

%end gataj36
```

MATLAB file: pertaj36.m

```
function [Texper]=pertaj36(Tfem,uncert,notrand)
%Purpose:
%         PERTK36 perturbs the mean value Texp(experimental temperature)
%         at a given uncertainty(standard deviation/mean value)
%Syntax:
%        [Texper]=pert(Tfem,uncert,notrand)
%Description:
%        Input:
%        * Tfem is vector with FEM temperatures
%        * uncert is the uncertainty to be considered
%        * notrand specifies probability n.  Scaled from 0 to 1 of nmax
%        Output:
%        * Texp is the perturbed Tfem at a given uncertainty and
probability.
%Remarks:
%        * If notrand==0, the a uniformly distributed random number is
chosen.

%Calculating the standard deviation.
stdev=uncert*Tfem;

%Calculating nmax.
nmax(1)=1/sqrt(2*pi)/stdev(1);
nmax(2)=1/sqrt(2*pi)/stdev(2);
nmax(3)=1/sqrt(2*pi)/stdev(3);

%Calculating n.
if notrand>0
   n=nmax*notrand;
elseif notrand==0
   n(1)=nmax(1)*rand;
   n(2)=nmax(2)*rand;
   n(3)=nmax(3)*rand;
end

%Checking that n is not equal to 0.
if n(1)==0
  n(1)=eps;
elseif n(2)==0
  n(2)=eps;
elseif n(3)==0
  n(3)=eps;
end
```

```
%Calculating Texper.
   Texperf(1)=stdev(1).^2*(1/stdev(1).^2*Tfem(1)+sqrt(-
2*log(n(1)*stdev(1)*sqrt(2*pi)))/stdev(1));
   Texperf(2)=stdev(2).^2*(1/stdev(2).^2*Tfem(2)+sqrt(-
2*log(n(2)*stdev(2)*sqrt(2*pi)))/stdev(2));
   Texperf(3)=stdev(3).^2*(1/stdev(3).^2*Tfem(3)+sqrt(-
2*log(n(3)*stdev(3)*sqrt(2*pi)))/stdev(3));

   Texperg(1)=stdev(1).^2*(1/stdev(1).^2*Tfem(1)-sqrt(-
2*log(n(1)*stdev(1)*sqrt(2*pi)))/stdev(1));
   Texperg(2)=stdev(2).^2*(1/stdev(2).^2*Tfem(2)-sqrt(-
2*log(n(2)*stdev(2)*sqrt(2*pi)))/stdev(2));
   Texperg(3)=stdev(3).^2*(1/stdev(3).^2*Tfem(3)-sqrt(-
2*log(n(3)*stdev(3)*sqrt(2*pi)))/stdev(3));

   Texper(1)=Texperf(1)+rand*(Texperg(1)-Texperf(1));
   Texper(2)=Texperf(2)+rand*(Texperg(2)-Texperf(2));
   Texper(3)=Texperf(3)+rand*(Texperg(3)-Texperf(3));


%end pertaj36
```

MATLAB file: three2six.m

```
function [Aout]=three2six(Ain)

Aout(:,1)=Ain(:,1);
Aout(:,2)=Ain(:,1);
Aout(:,3)=Ain(:,2);
Aout(:,4)=Ain(:,1);
Aout(:,5)=Ain(:,1);
Aout(:,6)=Ain(:,3);
```

MATLAB file: creapopaj36.m

```
function [A]=creapopaj36(Aguess,popnum)
%Purpose:
%        CREAPOPAJ36 creates a population of parameters aj, a3 & a6.
%Syntax:
%        [A]=creapopaj36(Aguess,popnum)
%Description:
%        Input:
%        * Aguess is a vector containing a guess value for aj, a3 and a6.
%        * popnum is the number of populations to be evaluated.
%        Output:
%        * A is a matrix.  Each row represents a combination of aj, a3 and
a6.
%Remarks:
%        * A safefactor is used in this function to force band limits.
%          Change this local variable as necessary.
%          Consider to increase pmutate and genum.
%          Default value is .2.

safefactor=.2; %Forcing band limits.
row=popnum;col=length(Aguess);

%Creating random numbers(from 0 to 1).
Ar=rand(row,col)*safefactor;

%Creating random negative values.
A1=ones(row,col);
for i=1:row
   for j=1:col
      if rand>.5
         A1(i,j)=-1;
      end
   end
end

%Creating matrix for mean and random deviation values.
for i=1:popnum
   Aw(i,:)=A1(i,:).*Ar(i,:).*Aguess(1,:);
   Au(i,:)=Aguess(1,:);
end

%Creating population by adding mean and random deviation values.
A=Au+Aw;

%end creapopaj36
```

MATLAB file: fitness.m

```
function [FIT]=fitness(Tan,Texper)
%Purpose:
%        FITNESS calculates the temperature difference |Texper - Tan|
%        at the selected location.
%Syntax:
%        [FIT]=fitness(Tan,Texper)
%Description:
%        Input:
%        * Tan is a vector containing the calculated analytical
temperature.
%        * Texper is the a vector containing the experimental temperature.
%        Output:
%        * FIT is a vector containing the absolute value of the
%          temperature diffences(Texper-Tanalyt) at the locations
%          where the Texper were measured.
%
%See also pertaj36, attemp6

%Checking vector length for Texper and Tan.
if size(Texper)~=size(Tan)
   'Texp size is not the same as Tan'
   break
end

%Calculating |Texper-Tan|
FIT=abs(Texper-Tan);

%end fitness
```

MATLAB file: fitnessaj36.m

```
function [fit]=fitnessaj36(FIT,gen)
%Purpose:
%        FITNESSAJ36 calculates the fitness of each individual in the
population.
%Syntax:
%        [FIT]=fitness(Tan,Texper)
%Description:
%        Input:
%        * Tan is a vector containing the calculated analytical
temperature.
%        * Texper is the a vector containing the experimental temperature.
%        Output:
%        * FIT is a vector containing the absolute value of the
%          temperature diffences(Texper-Tanalyt) at the locations
%          where the Texper were measured.
%
%See also pertaj36, attemp6


popnum=length(FIT);

sump1=0;
sump2=0;
sump3=0;

%Weighting equally all chromosomes.
for i=1:popnum
   sump1=(FIT{i}(1,1))+sump1;
   sump2=(FIT{i}(1,2))+sump2;
   sump3=(FIT{i}(1,3))+sump3;
end

for i=1:popnum
   col1(i)=FIT{i}(1,1)/sump1;
   col2(i)=FIT{i}(1,2)/sump2;
   col3(i)=FIT{i}(1,3)/sump3;
end

%Adding weighted chromosomes for each individual.
sump=(col1+col2+col3);

%Calculating fitness.
fit=sump/3;
```

```
%end fitnessaj36
```

MATLAB file: geninfoaj36.m

```
function [BESTAj,BESTA3,BESTA6,STATS]=geninfoaj36(FIT,A)
%Purpose:
%        GENINFOAJ36 keeps track of the fittest individual of population.
%Syntax:
%       [BESTAj,BESTA3,BESTA6,STATS]=geninfoaj36(FIT,A)
%Description:
%       Input:
%       * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of chromosomes aj,
a3 & a6.
%       * K is a matrix.  Each row represents a combination of aj, a3 and
k6.
%       Output:
%       * STATS=[ minfit meanfit maxfit stdevfit].
%       * BESTK3 & BESTK6 are corresponding chromosomes aj, a3 and a6 of
fittest individual.
%
%See also fitnessaj36, creapopaj36, mutateaj36

%Statistical values.
[minfit,bestind]=min(FIT);
meanfit=mean(FIT);
maxfit=max(FIT);
stdevfit=std(FIT);
STATS=[minfit meanfit maxfit stdevfit];

%Declaring chromosomes of best individual.
BESTAj=A(bestind,1);
BESTA3=A(bestind,2);
BESTA6=A(bestind,3);

%end geninfoaj36
```

MATLAB file: mateaj36.m

```
function [MATE2,GLOBFIT]=mateaj36(FIT)
%Purpose:
%        MATEAJ36 mates individuals among population by using a roulette
technique.
%Synntax:
%        [MATE2,GLOBFIT]=mateaj36(FIT)
%Description:
```

```
%        Input:
%        * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of aj, a3 & a6.
%        Output:
%        * MATE2 is a vector containing the row number of mate2.
%        * GLOBFIT is a matrix containing the fitness for each
corresponding
%          combination of aj, a3 & a6 based on FIT.
%
%See also fitnessaj36

popnum=size(FIT,2);

%Assigning higher mating probability for small values of FIT.
invfit=FIT.^(-1);
GLOBFIT=invfit/sum(invfit);

%Assigning mate2 to mate1 by using the roulette technique.
for mate1=1:popnum
   rolwheel=rand; %Spinning the roulette./Random hit.
   flag=1;
   selMATE2=0;

   %Finding corresponding mate2 for given mate1
   while flag==1
      selMATE2=selMATE2+1;
      rolocat=sum(GLOBFIT(1:selMATE2));

      if rolocat>=rolwheel
         MATE2(mate1)=selMATE2;
         flag=0;
      end

   end

end

%end mateaj36
```

MATLAB file: xoveraj36.m

```
function[Aout]=xoveraj36(MATE2,FIT,GLOBFIT,Texper,X_LOC,TIME,numev6,q,L,K,
Ain)
%Purpose:
%       XOVERAJ36 crosslinks information between parents in order to
create
%       offsprings.  After mating, the genetic information passed to the
%       offsprings can be done in the following manners: genetic info
%       from parent1, parent2 or the mean value from parent1 & parent2.
%Syntax:
%
[Aout]=xoveraj36(MATE2,FIT,GLOBFIT,Texper,X_LOC,TIME,numev6,q,L,K,Ain)
```

```
%Description:
%       Input:
%       * MATE2 is a vector containing the row number of mate2.
%       * FIT is a matrix containing the absolute value of the
%          temperature differences(Texper-Tanalyt) at the locations
%          where the Texper were measured.  Each row corresponds to
%          results obtained using a given combination of aj, a3 & a6.
%       * GLOBFIT is a matrix containing the fitness for each
corresponding
%         combination of aj, a3 & a6 based on FIT.
%       * Texper is the a vector containing the experimental temperature.
%       * X_LOC is a vector containing the locations where Texper were
measured.
%       * q is the magnitude of heat flux.
%       * L is a vector containing the thickness values of all 6 layer.
%       * Ain is a matrix containing the population. Each row is an
individual made of aj, a3 & a6.
%       Output:
%       * Aout is a matrix containing the cross-linked population.
%
%See also mateaj36,fitnessaj36

popnum=length(FIT);

for mate1=1:popnum

   %Creating a mean value between parent 1 and parent 2.
   Aout(mate1,:)=(Ain(mate1,:).*GLOBFIT(mate1)...
   +Ain(MATE2(mate1),:).*GLOBFIT(MATE2(mate1)))./...
   (GLOBFIT(mate1)+GLOBFIT(MATE2(mate1)));

   %Calculating |Texper-Tan|
   [A]=three2six(Aout(mate1,:));
   [Tout]=attemp6(X_LOC,TIME,numev6,q,L,K,A);
   [FITout]=fitness(Tout,Texper);

   %Cross-linking genetic information directly from parent 1,
   %parent 2, or the mean value between parent 1 & parent 2.
   if FITout<=FIT(mate1)& FITout<=FIT(MATE2(mate1))
      dummy=1;
   elseif FIT(mate1)<FITout & FIT(mate1)<FIT(MATE2(mate1)) & rand<.95
      Aout(mate1,:)=Ain(mate1,:);
   elseif FIT(MATE2(mate1))<FITout & FIT(MATE2(mate1))<FIT(mate1) &
rand<.95
      Aout(mate1,:)=Ain(MATE2(mate1),:);
   end

end

%end xoveraj36
```

MATLAB file: mutateaj36.m

```
function [A]=mutateaj36(A,pmutate)
```

```
%Purpose:
%       MUTATEAJ36 mutates one or both of the chromosomes of an
individual.
%Syntax:
%       [Aout]=mutateaj36(Ain,pmutate)
%Description:
%       Input:
%       * Ain is a matrix containing the population.
%         Each row is an individual made of chromosomes aj, a3 & a6.
%       * pmutate is the probability of mutation.
%       Output:
%       * Aout is a matrix containing the mutated Kin
%
%See also xoveraj36

popnum=size(A,1);
laynum=size(A,2);
safefactor=.5; %Defining maximum magnitude of mutation.

%Checking on mutation./Executing mutation.
for i=1:popnum

   for j=1:laynum
      random=rand;
      randsign=rand;

      if  random<=pmutate & randsign<=.5
         A(i,j)=A(i,j)*(1+randsign*safefactor);
      elseif random<=pmutate & randsign>.5
         A(i,j)=A(i,j)*(1-randsign*safefactor);
      end

   end

end

%end mutateaj36
```

Vita

Joel Barreto was born on March 1, 1972 in Ponce, Puerto Rico. Upon graduation from high school he moved to the United States to pursue his college education. He enrolled at The Ohio State University (OSU), receiving a Bachelor in Science degree in Mechanical Engineering in June, 1996. During his undergraduate studies he worked as a researcher at OSU and as an applications engineer for Corning Incorporated.

In August 1996, he began his graduate studies at Virginia Polytechnic Institute and State University where he joined the Thermal Radiation Group. He became a fellow of the National Consortium for Graduate Minority Degrees in Engineering and Science. During his graduate studies he worked as a Graduate Research Assistant. In July 1998, he completed the requirements for a Master of Science degree in Mechanical Engineering.

Joel Barreto