Fall 1998

An interactive system for the estimation of emissivity of a wafer in a rapid thermal processing chamber

Maurizio Fulco
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

AN INTERACTIVE SYSTEM FOR THE ESTIMATION OF EMISSIVITY OF A WAFER IN A RAPID THERMAL PROCESSING CHAMBER

by
Maurizio Fulco

Rapid thermal processing (RTP) is a method of thermally processing wafers for the manufacture of integrated circuits. During the thermal processing of wafers, it is essential that the wafer temperature follow a pre-specified temperature trajectory and that the temperature across the wafer be uniform especially at high temperatures. To ensure that the above objectives of RTP temperature control be met at any time during the process, the estimation of some parameters of the process is of fundamental importance in the design of the control system.

This thesis demonstrates the implementation of an interactive software system in which the emissivity of wafers in a 3-zone RTP station can be estimated on-line. The simulation of the RTP system is performed to ensure the proper performance of the estimator and the closed loop control system. In addition, it is necessary for the control of temperature uniformity of the wafer to implement simulations of the control system and to experiment with new ways to obtain states and parameters estimations.

The implementation of the system is carried out on a Pentium based PC using LabVIEW and G Math Toolkit with the full advantages of graphical programming or G. The capabilities of LabVIEW to directly interface with the system using the data acquisition boards provided motivates the utilization of this software system.

AN INTERACTIVE SYSTEM FOR THE ESTIMATION OF EMISSIVITY OF A WAFER IN A RAPID THERMAL PROCESSING CHAMBER

by
Maurizio Fulco

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 1999

APPROVAL PAGE

AN INTERACTIVE SYSTEM FOR THE ESTIMATION OF EMISSIVITY OF A WAFER IN A RAPID THERMAL PROCESSING CHAMBER

Maurizio Fulco

Dr. Onofrio L. Russo, Thesis Advisor
Associate Professor of Physics, NJIT

Date

Dr. Bernard Friedland
Distinguished Professor of Electrical and Computer Engineering, NJIT

Date

Dr. Edwin Cohen
Professor of Electrical Engineering, NJIT

Date

Dr. Sergey Belikov
Candescent Technologies

Date

BIOGRAPHICAL SKETCH

Author: Maurizio Fulco

Degree: Master of Science

Date: January 1999

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology,
  Newark, NJ, 1999

- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology,
  Newark, NJ, 1996

Major: Electrical Engineering

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Dr. Onofrio L. Russo, not only for serving as my thesis advisor, but also for giving me encouragement and reassurance during difficult periods. The same is valid for my family. Without your support this would not have happened. I also express my appreciation to Dr. Sergey Belikov on whose theoretical work much of this thesis is based on.

I would also like to thank Dr. Bernard Friedland for his guidance and support over the years and Dr. Edwin Cohen for serving as members of the committee. Finally, I wish to thank Vincenzo Pappano for the long hours spent in the lab assisting me when difficulties where encountered and also for providing insights to some of the work on this thesis.

This investigation was supported by the National Science Foundation under the grant ECS-9312451.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Rapid Thermal Processing

Rapid Thermal Processing (RTP) is an advanced method of semiconductor manufacturing in which silicon wafers undergo a number of processing steps quickly, typically at high temperatures. In the RTP method, a single wafer is processed separately in a single, small chamber. High levels of radiation from a lamp array are applied directly to the wafer, eventually rising its temperature to high values. The wafer is then kept at a high temperature for the amount of time needed for processing. When the thermal process is complete, the wafer is cooled down. The control of the pre-specified process temperature trajectory and the control of the temperature uniformity across the surface of the wafer are the main issues investigated by researchers in order to improve the technology of RTP. A temperature uniformity of 1K is a typical performance goal. The improvements will advance the state-of-the-art in semiconductor manufacturing. [1]

The RTP method promises process-related improvements over conventional batch processing methods. In conventional methods, many wafers are simultaneously treated thermally in large temperature controlled furnaces. The thermal processing of the wafers with this procedure is slow, usually in terms of minutes or even hours. Speed is limited by the large thermal masses of the oven walls, which increases the thermal cost of the process. The processing step such as oxidation occurs only at high temperatures. At such high temperatures, the problem of diffusion and thermal shock becomes an important factor. Therefore, the length of time at which the wafers are subjected to high

temperatures presents a problem as the circuit scaling continues to shrink as is the current trend in the semiconductor industry.

Some advantages of rapid thermal processing of the wafer are the reduction of the processing time, which implies reduction of the overall thermal cost of processing a wafer. The thermal mass of the RTP chamber is reduced due to the smaller dimensions of the chamber relative to the conventional oven and also due to the fact that the walls of the reaction chamber are water-cooled and remain at room temperature. Another advantage of rapid thermal processing is the inherent flexibility associated with the method. By being able to specify the temperature trajectory in time, many different types of processing can be accomplished by a single RTP station. This flexibility makes the RTP method more suitable for the programmable factory environment. A diagram of an RTP chamber is given in figure 1.1.

During rapid thermal processing it is essential that the RTP station has the ability to follow a given process temperature trajectory, and to maintain uniformity of the temperature across the surface of the wafer while following the process temperature trajectory. The close proximity of the wafer being processed to the heat source, and the inherent lack of thermal stability inside the processing chamber makes the uniformity issue a difficult one to solve. The problem of controlling RTP systems by using different methods has been studied in previous researches with satisfactory results [1],[2].

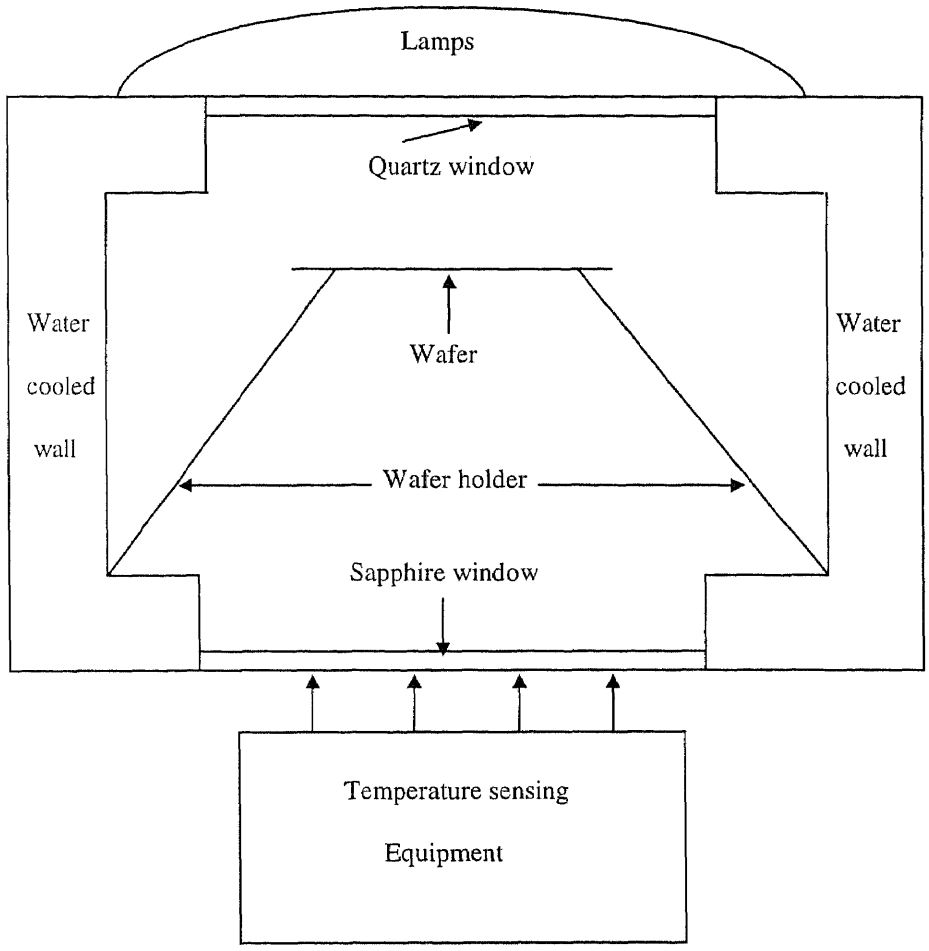


Figure 1.1 Diagram of an RTP chamber.

1.2 Description of the RTP System

A typical axi-symmetrical RTP system comprises a cylindrical reaction chamber housing a circular semiconductor wafer and a bank of high-power tungsten-quartz lamps which are used to heat the wafer by infrared radiation. The lamps, which are arranged in concentric rings, are positioned above a quartz window over the reaction chamber. Each ring of lamps is independently controlled to achieve the desired temperature profile.

The system under investigation at NJIT is an experimental 3-zone TI-designed RTP system. It has three concentric rings, with 1, 12, and 24 lamps, respectively. The

rated power of the central lamp is two kW whereas the lamps in the remaining two rings are one kW each. The semiconductor wafer is positioned 0.085" below a quartz window and is supported by a three-pin wafer holder. The stainless-steel walls of the RTP chamber are water-cooled to provide a constant temperature of approximately 300 K. The quartz window at the top of the chamber is water-cooled at the edges. In addition, the quartz window is cooled by room temperature air being blown over its top surface. The ambient in the process chamber may be vacuum or a process gas, depending on the thermal process being performed.

In the system under investigation, the temperature of the wafer is sensed by means of thermocouples mounted along a radial line of the wafer for experimental purposes only. The processed output of the sensors are input to the control computer which implements the controller and the observer to produce a set of independent control commands to the lamp rings. The chamber has also provisions for a remote infrared temperature sensor through a sapphire window at the bottom. As a preferred alternative, the radiative energy of the wafer can be measured to monitor its temperature using a non-contacting device such as a multi-wavelength imaging pyrometer.

1.3 Necessity for Estimation of Emissivity

A challenge in RTP temperature control is the accurate measurement of the wafer temperature at a multitude of points across the surface. The most widely used techniques for temperature measurements in RTP systems are thermocouples and pyrometers [20]. Thermocouples are used in experiments to develop and test feedback control algorithms and calibrate pyrometers. However, their use in production line is not realistic since

thermocouples must maintain contact with the wafer being processed. By contrast, pyrometers are contact-less devices which have the ability to measure the temperature of the wafer without touching it. A problem arises when using pyrometers. Pyrometers do not measure the temperature directly but they measure the intensity of the thermal radiation emitted from a particular point of the wafer. In this case, the emissivity of the wafer being processed is needed to convert from radiant power to temperature.

According to Planck's radiation theory, the radiation from the heated wafer that reaches a pyrometer is given by

$$W(T) = \varepsilon(\lambda, T)\gamma^s d\,\text{Planck}(\lambda_0, T) \tag{1.1}$$

where $\varepsilon(\lambda,T)$ is the spectral emissivity of the wafer, γ^s is the gain of the pyrometer, d is the width of the optical filter, λ_o is the central wavelength of the optical filter, and Planck(λ_o,T) is the Planck function for black-body radiation.

Spectral emissivity, $\varepsilon(\lambda)$, is defined as the ratio of the radiation emitted by a wafer at a given wavelength, angle of incidence, and plane of polarization to that emitted from a black body under the same conditions. The spectral emissivity of an object in thermal equilibrium at a given wavelength, angle of incidence, and plane of polarization, is identical to its absorptivity, $a(\lambda)$, for the same conditions. The emissivity of a silicon wafer changes as the processing temperature of the wafer changes. Also, the emissivity of a wafer will have large variations if the wafer has surface coatings, such as poly-silicon film deposited over a dielectric layer. These variations can introduce serious difficulties in conventional pyrometry. Furthermore, other system components such as stray radiation from the lamps can affect the pyrometer signal, but the effect of stray lamp radiation can

be minimized by various filtering methods. However, the spectral emissivity of the wafer must be known, if using pyrometers, to account for deviations from black body behavior. Unknown wafer emissivity remains a problem.

Spectral emissivity should not be set at the pyrometer wavelength, λ_p. If so done, an incorrect temperature measurement given by the following expression will be introduced [21]

$$\frac{1}{T_a} = \frac{1}{T_r} + \frac{\lambda_p}{c_2} \ln\left(\frac{\varepsilon_a}{\varepsilon_r}\right) \tag{1.2}$$

where T_a is the measured temperature in K, T_r is the real temperature in K, c_2 is Planck's second radiation constant, ε_a is the emissivity value incorrectly set at the pyrometer wavelength, λ_p, and ε_r is the real emissivity value. An example will show the error introduced if the emissivity was incorrectly set at the pyrometer wavelength. The spectral emissivity of a pure silicon wafer at 800°C and at a wavelength of 1μm has a value of 0.662. If a layer of oxide of thickness of 0.2μm covered by a layer of poly-silicon of the same thickness is deposited on the wafer the spectral emissivity of the wafer becomes 0.178. A pyrometer calibrated on pure silicon would measure a temperature of 704°C for the coated wafer instead of the actual temperature of 800°C. The error would be larger if the pyrometer wavelength falls in a range where the spectral emissivity changes rapidly with film thickness. The above discussion clearly demonstrates that finding algorithms and methods to estimate the emissivity on line during the process is essential to account for wafer-to-wafer variations.

The inverse function of (1.1) is

$$T(W) = \frac{14387.2}{\lambda_0 \log\left(1 + \frac{3.74173 \times 10^{14} \varepsilon \gamma^s d}{\lambda_0^{\,5} W}\right)} \tag{1.3}$$

where T is the temperature in K of the spot on the wafer where the pyrometer is focused, W is the output of the pyrometer in volts corresponding to this temperature, λ_0 and d are in μm, and ε is the emissivity estimated in real time.

The objective of the present research is to propose an interactive software system which demonstrates that the emissivity of wafers in a RTP chamber can be estimated for different temperatures and wavelengths. The parameter's emissivity estimation is implemented by the persistent excitation of the radiation from the lamps and the introduction of a dynamic observer.

1.4 Research Objective

The objectives of this thesis are to generally describe the physical models and to demonstrate the computer interfacing and the control of the interactive system. Several general useful emissivity models are suggested and presented. The nature of these models is such that, although they may describe the correct trend in typical situations, the parameters of the models are almost never known with sufficient accuracy. In this situation, the application of a non-linear observer is justified. An interactive computer system when connected to an RTP station gives one the freedom to experiment with different models and observers. The system is designed so as to collect real time measurements and use the data for a quick assessment and analysis of the models. It also provides on-line information about appropriate emissivity models derived from physical

theories and situations where these models may be used. In addition, the State Dependent Riccati Equation Filter (SDREF) derived elsewhere is applied to this system to estimate states and disturbance parameters [18], [19]. The simulation results are shown in this work.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Introduction

This chapter presents a brief summary of the wavelength dependent emissivity models used in the demonstration of the interactive software system as well as the dynamic model of the RTP station used in the simulation. The model based adaptive controller is also presented since it was used in the SDREF simulations. The adaptive control law was derived in [2].

2.2 Wavelength Dependent Emissivity Models

RTP processing for semiconductor wafers occur at temperatures low enough so that the contributions to the emissivity at short wavelengths (less than 1.0 μm) is negligible. Two physical models for λ greater than 1.0 μm were selected. For this range of wavelength and under certain conditions (e.g. lightly doped Si), the emissivity is a sensitive function of wavelength λ, and also of temperature T. The dominant mechanism for the dependence of the emissivity, ε, on wavelength is that due to free carrier absorption for photon energies below the band gap. The region between 1 - 3.5 μm indicates an absorption coefficient for Si which is proportional to $\lambda^{1.5}$[3], and to λ^2 for wavelengths above 5.0 μm [4]. The absorption properties and the corresponding emissivity of Si as a function of temperature and wavelength have been detailed elsewhere in a rather extensive review [5].

Two emissivity models both depending on the parameters p, wavelength λ, and temperature T, have been selected as a basis for this demonstration. One consists of (a) an emissivity function which is a non-linear function of the parameters, p, [6] and the other [6] of (b) an emissivity function which is a linear function of these same parameters, p.

In case (a), the emissivity function can be separated into two distinct regions. The first region is that in which the function varies as $\lambda^{1.5}$ for values of λ from 1.0 μm to λ_o and the next region is that in which the function varies as λ^2 for values of $\lambda > \lambda_o$. This is formalized in the non-linear (with respect to the parameter, p) case as

$$\varepsilon(\lambda,T) \approx f(\mathbf{p},\lambda,T) \tag{2.1}$$

where $\mathbf{p} = (p_1,\ldots,p_n)$ is an n-dimensional parameter $n = 1,2,3$, and

$$f(\mathbf{p},\lambda,T) = \bar{f}_1(\mathbf{p}_1,\lambda,T) + \bar{f}_2(\mathbf{p}_2,\lambda,T) \tag{2.2}$$

where

$$\bar{f}_1(\mathbf{p}_1,\lambda,T) = \begin{cases} f_1(\mathbf{p}_1,\lambda,T), \lambda \le \lambda_0 \\ 0, \lambda > \lambda_o \end{cases} \tag{2.3}$$

and

$$\bar{f}_2(\mathbf{p}_2,\lambda,T) = \begin{cases} 0, \lambda \le \lambda_0 \\ f_2(\mathbf{p}_2,\lambda,T), \lambda > \lambda_0 \end{cases} \tag{2.4}$$

In case (b), which applies for the conditions where the functions are linear with respect to the parameters, p, we can write

$$\varepsilon(\lambda,T) \approx \sum_{j=1}^{n} p_j f_j(\lambda,T) \tag{2.5}$$

as the model for the region for λ>1.0 μm. Theoretical and experimental studies using Fourier Transform Infrared Spectroscopy [4], a special mathematical treatment of

experimentally measured spectral data, show that a valid approximation for the wafer emissivity as a function of wavelength in the working interval [1, 25μm] is

$$\varepsilon(\lambda,T)=\sum_{l_{\min}}^{l_{\max}} p_l(T)\lambda^l \tag{2.6}$$

where $l_{\min}$=0,-1,-2, and $l_{\max}$=0,1,2.

The dynamic observer depends on the model that is selected.

2.3 Heat Transfer Model for the Wavelength-dependent Emissivity of a Wafer

In a manner similar to [7], the differential equation for the temperature on the wafer with emissivity model (2.5) in an axi-symmetrical RTP chamber is

$$\dot{T}=\sum_{j=1}^{n} p_j F_j(T)+2h^{-1}[S+G(\mathbf{U})] \tag{2.7}$$

where, from radiation heat transfer theory as applied to an opaque diffusive body like silicon at high temperatures, the function $F_j(T)$ is calculated by the formula

$$F_j(T)=-2h^{-1}\int_0^{\infty} f_j(\lambda,T)\text{Planck}(\lambda,T)d\lambda/(\rho c(T)) \tag{2.8}$$

where ρ is the density, $c(T)$ is the specific heat of silicon, h is the thickness of the wafer, S is a parameter which accounts for convection and environmental heat flux, and $G(\mathbf{U})$ is the heat flux generated by the vector $\mathbf{U}$ (control voltages applied to the lamp rings). Planck(λ,T) is the Planck function for black-body radiation given by

$$\text{Planck}(\lambda,T)=\frac{4\pi^2\hbar c^2}{\lambda^5}\frac{1}{\exp(2\pi\hbar c/kT\lambda)-1} \tag{2.9}$$

where $\hbar$ is Planck's constant, k is Boltzmann's constant, and c is the speed of light. It is necessary that the integral $\int_0^\infty f_j(\lambda,T)\,\text{Planck}(\lambda,T)d\lambda$ be calculated for solving the heat transfer equation (2.7). Using the experimental finding - eq. (2.6)- instead of the more general case of equation (2.5), this translates to evaluating the integral $\int_0^\infty \lambda^l\,\text{Planck}(\lambda,T)d\lambda$, l=-2, -1, 0, 1, 2. The above integral can be evaluated analytically [15]. The result of the integral is $\alpha_l T^{4-l}$.

The emission function $E(T)$, defined as the heat flux emitted via radiation from the unit area of the wafer is calculated by the formula

$$E(T)=\int_0^\infty \varepsilon(\lambda,T)\text{Planck}(\lambda,T)d\lambda\ . \tag{2.10}$$

If the emissivity model is characterized by equation (2.6), the value for the emission function after the calculation of the integral above is

$$E(T)=\sum_{l_{\min}}^{l_{\max}} \alpha_l p_l(T) T^{4-l} \tag{2.11}$$

where the α_l values were calculated in [15], and σ is Stefan-Boltzman constant

$$\begin{aligned}\sigma=\alpha_0&=5.67\times10^{-8}[W/(m^2K^4)]\\ \alpha_1&=3.02\times10^{-10}[W/(mK^3)]\\ \alpha_2&=2.97\times10^{-12}[W/K^2]\\ \alpha_{-1}&=1.51\times10^{-5}[W/(m^3K^5)]\\ \alpha_{-2}&=5.15\times10^{-3}[W/(m^4K^\wedge)]\end{aligned}$$

The dynamic model for RTP temperature control of a thin silicon wafer in an axi-symmetric system with three rings of heating lamps using the above emission function

calculated using the emissivity model (2.6) is described by the following partial differential equation

$$\rho c(T(r,t))\frac{\partial T(r,t)}{\partial t} = k(T(r,t))\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T(r,t)}{\partial r}\right) - 2h^{-1}E(T(r,t)) + 2h^{-1}\sum_{l=1}^{3}[G_l(r)u_l(t) + S(r,t)] \tag{2.11}$$

$$T(r,0) = T_0(r),\ 0 \le r < R \tag{2.12}$$

$$[\partial T(r,t)/\partial r]_{r=R} = 0 \tag{2.13}$$

where R is the radius of the wafer, h is the wafer thickness, r is the radial position along the wafer, ρ is the density of the wafer material, $T(r,t)$ is the temperature of the wafer at point r at time t, $c(T(r,t))$ is the specific heat of the wafer, and $k(T(r,t))$ is the thermal conductivity of the wafer. The above partial differential equation (2.11) incorporates:

1. Nonlinear lamp ring radiation functions G_l relating the voltage U in the range 0-5 Volts supplied to the lth ring of heating lamps to the heat power reaching the surface of the wafer.

2. The heat flux $S(r,t)$ to and from the edge of the wafer representing convection and not modelled environmental and disturbance flux.

3. The control function $u_l(t)$ which represents the power emitted by the lth ring of heating lamps. It is evident that for realistic control this function must be set ≥ 0.

For the emissivity model (2.1), the equation is

$$\dot{T} = F(\mathbf{p},T) + 2h^{-1}[S + G(\mathbf{U})] \tag{2.14}$$

where

$$F(\mathbf{p},T) = -2h^{-1}\int_0^{\infty} f(\mathbf{p},\lambda,T)d\lambda / (\rho c(T)) \tag{2.15}$$

2.4 RTP Dynamic Model

The dynamic model of the RTP system used to implement the RTP simulator, controller, and observer exploits the circular symmetry of the physical configuration where the state variables are the coefficients of a Bessel function expansion. Parameters of the model are derived from experimental data obtained from the actual RTP system.

An investigation reported in [15] has shown by finite element analysis that the temperature distribution in a semiconductor wafer $T(r,t)$ can be represented by the following Bessel function

$$T(r,t) = \bar{T}(t) + x_1(t) + x_2(t)J_0(\mu_1 r / R) + x_3(t)J_0(\mu_2 r / R) \tag{2.16}$$

where $J_0(x)$ is the Bessel function of the first kind of zero order, μ_1=3.8317,and μ_2=7.0156 are solutions of the equation $dJ_0(\mu)/d\mu = 0$, and $\bar{T}(t)$ is the reference temperature. The coefficient $x_1(t)$ is the average deviation from the reference uniform temperature profile.

The coefficients $x_n(t)$, n = 1,2,3, of the Bessel expansion of equation 2.16 are differences between the temperature of the wafer and the reference temperature. They also represent the state variables of the system and satisfy the following state ordinary differential equations:

$$\begin{aligned}
\dot{x}_1 &= \dot{\bar{T}} + \frac{1}{\rho c(\bar{T}(t)+x_1)}\left[-F(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^4 + P_1\right] \\
\dot{x}_2 &= \frac{1}{\rho c(\bar{T}(t)+x_1)}\left[\left(-(\mu_1/R)^2 k(\bar{T}(t)+x_1) - 4F(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^3\right)x_2 + P_2\right] \\
\dot{x}_3 &= \frac{1}{\rho c(\bar{T}(t)+x_1)}\left[\left(-(\mu_2/R)^2 k(\bar{T}(t)+x_1) - 4F(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^3\right)x_3 + P_3\right]
\end{aligned} \tag{2.17}$$

where

$$P_n = 2h^{-1}\left[\sum_{l=1}^{3} G_{ln}(U_l) + S_n(t)\right], \quad n = 1,2,3 \tag{2.18}$$

and $F(x) = 2h^{-1}\sigma\varepsilon(x)$.

The simulation runs were conducted for the silicon wafer of R = 76.2 *mm* radius, h = 0.635 *mm* thickness, and ρ = 2330 kg/m^3. Three thermocouples are used for output temperature measurements at three different points along a radial line, r_1=0, r_2= 23 *mm*, and r_3= 46 *mm*. The temperature dependent specific heat $c(T)$ and thermal conductivity $k(T)$ are given in Table 1, and the temperature dependent emissivity is given in Table 2.

Table 2.1 Temperature Dependent Specific Heat c and Thermal Conductivity k for Silicon.

T[K]	200	400	600	800	1000	1200
C[J/kg·K)]	549	780	856	900	934	955
k[W/(m·K)]	264	98.9	61.2	42.2	31.2	25.7

Table 2.2 Temperature Dependent Emissivity for Silicon.

T[K]	300	550	700	740	1000
ε	0.2	0.2	0.5	0.7	0.7

The coefficients G_{ln} were derived in an identification experiment reported in [17]. According to this work these estimated values are the following:

$$G_{11} = 2808$$

$$G_{12} = -1.22$$

$$G_{13} = 3908$$

$$G_{21} = 17043$$

$$G_{22} = 2480$$

$$G_{23} = -3254$$

$$G_{31} = 20731$$

$$G_{32} = -615$$

$$G_{33} = -2232$$

The above state equations can also be represented in matrix form, a notation which will become handy when the state-dependent Riccati equation technique is used to estimate the states and the parameters of the system. The representation of the system in matrix form is the following:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(x,t)\mathbf{x} + \mathbf{B}(x,t)\mathbf{u} + \mathbf{ES} \\ \mathbf{y} &= \mathbf{Cx} \end{aligned} \tag{2.19}$$

where the matrices **A**, and **B** are defined as follows

$$\mathbf{A}(x,t) = \begin{bmatrix} \frac{-F(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^4}{x_1\rho c(\bar{T}(t)+x_1)} & 0 & 0 \\ 0 & \frac{-(\frac{\mu_1}{R})^2 k(\bar{T}(t)+x_1)-4RF(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^3}{\rho c(\bar{T}(t)+x_1)} & 0 \\ 0 & 0 & \frac{-(\frac{\mu_2}{R})^2 k(\bar{T}(t)+x_1)-4F(\bar{T}(t)+x_1)(\bar{T}(t)+x_1)^3}{\rho c(\bar{T}(t)+x_1)} \end{bmatrix}$$

$$\mathbf{B}(x,t) = \begin{bmatrix} 2808b & 17043b & 20731b & -1 \\ -1.22b & 2480b & -615b & 0 \\ 3908b & -3254b & -2232b & 0 \end{bmatrix}$$

where the coefficient b is

$$b = \frac{2h^{-1}}{\rho c(\bar{T}(t) + x_1)}.$$

The matrices **C** and **E** are defined as follows

$$\mathbf{C}=\begin{bmatrix} 1 & J_0(\frac{\mu_1 r_1}{R}) & J_0(\frac{\mu_2 r_1}{R}) \\ 1 & J_0(\frac{\mu_1 r_2}{R}) & J_0(\frac{\mu_2 r_2}{R}) \\ 1 & J_0(\frac{\mu_1 r_3}{R}) & J_0(\frac{\mu_2 r_3}{R}) \end{bmatrix}$$

$$\mathbf{E}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It remains to define the vector **u** as

$$\mathbf{u}=\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dot{\bar{T}} \end{bmatrix}.$$

2.5 Adaptive Control Algorithm

An algorithm of an adaptive control system for this RTP station was developed and implemented in [2]. The same algorithm is used here and is presented for reference. The controller is needed for the generation of the persistent excitation necessary for the observer used to estimate the parameter emissivity and for the simulation of the state dependent Riccati equation filter presented later in this thesis for which the observed states and parameters are fed back into the controller. The control algorithm is also needed for the generation of the control functions that later can be converted to the control voltages to the lamp rings. The state variables and the disturbance parameter vector S are required for the algorithm.

The controller is designed according to the separation principle in which it is assumed that the state variables and the parameters are measurable or known a priori. Then, the unknown parameters are estimated by the available observers.

The feedback control law was derived using the method of feedback linearization to reduce the system in the form $\dot{x}_n = -\lambda_n x_n \quad , n = 1,2,3$ for given positive λ. The derivation of the feedback control is derived in [2]. There, the control algorithm is represented in the form $\mathbf{u} = \mathbf{G}^{-1}\,\mathbf{V}(t,\mathbf{x},\mathbf{S})$ where $\mathbf{V}$ is the following feedback control law

$$\mathbf{V}(t,\mathbf{x},\mathbf{S}) = \begin{bmatrix} -\lambda_1 x_1 \rho c\left(\overline{T}(t)+x_1\right) + \dot{\overline{T}}(t)\rho c\left(\overline{T}(t)+x_1\right) + F\left(\overline{T}(t)+x_1\right)\left(\overline{T}(t)+x_1\right) - S_1 \\ -\lambda_2 x_2 \rho c\left(\overline{T}(t)+x_1\right) + ((\mu_1/R)^2 k(\overline{T}(t)+x_1) + 4F\left(\overline{T}(t)+x_1\right)(\overline{T}(t)+x_1)^3)x_2 - S_2 \\ -\lambda_3 x_3 \rho c\left(\overline{T}(t)+x_1\right) + ((\mu_2/R)^2 k(\overline{T}(t)+x_1) + 4F\left(\overline{T}(t)+x_1\right)(\overline{T}(t)+x_1)^3)x_3 - S_3 \end{bmatrix}$$

derived from the state equations (2.17). The λ_i are the desired closed loop eigenvalues. The control vector $\mathbf{u}$ can be represented in the following linear form with respect to λ;

$$\mathbf{u}(t,\mathbf{x}) = \mathbf{a}_1(\mathbf{x})\lambda_1 + \mathbf{a}_2(\mathbf{x})\lambda_2 + \mathbf{a}_3(\mathbf{x})\lambda_3 + \mathbf{b}(\mathbf{x}) \qquad (2.20)$$

The control vector is admissible if and only if $\mathbf{0} \le \mathbf{u}(t,\mathbf{x}) \le \mathbf{U}^{max}$. For stability, the eigenvalues must be positive and their maximum values are limited by physical constraints. Eigenvalues were chosen to be equal and to have a maximum upper bound of 1 for convenience. Suitable values were obtained by linear programming method [2].

2.6 Simulation Results

A simulation was run to ensure proper operation of the controller using the RTP model described in paragraph 2.4 and the adaptive control law of paragraph 2.5 and assuming that the states are measured from the three thermocouple measurements and the

disturbance parameters are known. The observers of chapter 3 will be inserted in the closed loop control system to estimate the parameters that in these simulations are assumed known. Simulation graphs are presented in the following figures.

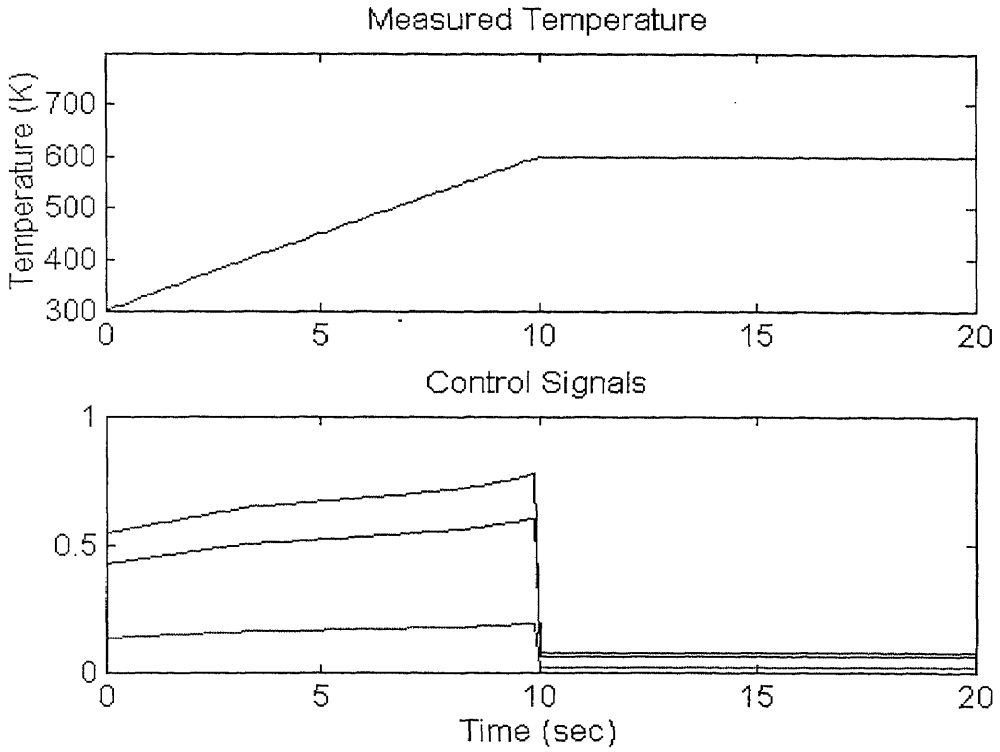


Figure 2.1 Open Loop Control Simulation Results

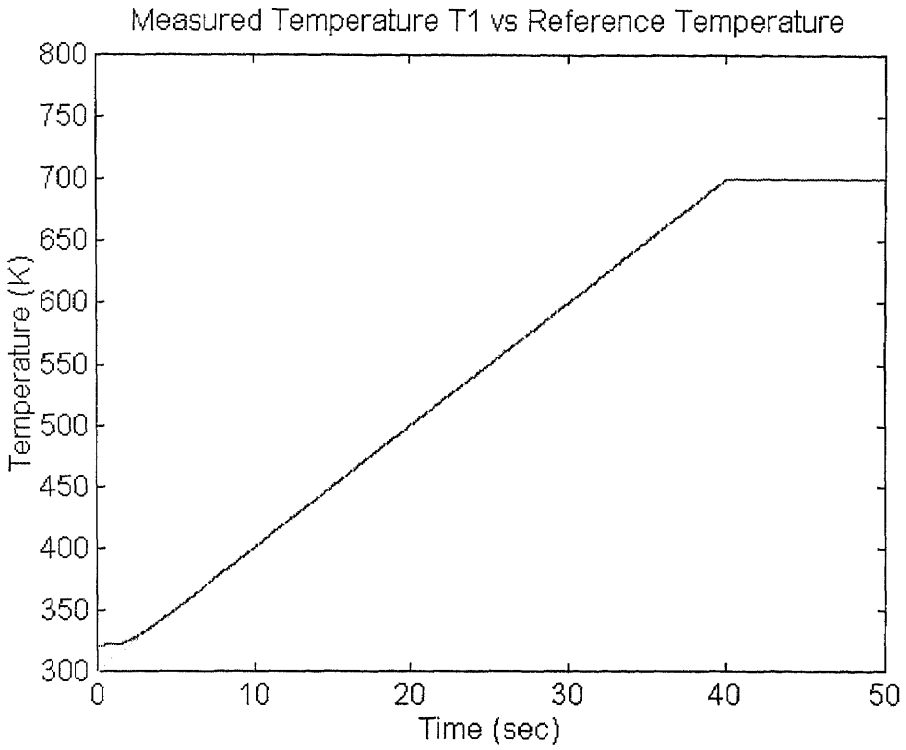


Figure 2.2 Closed Loop Temperature Tracking and Control

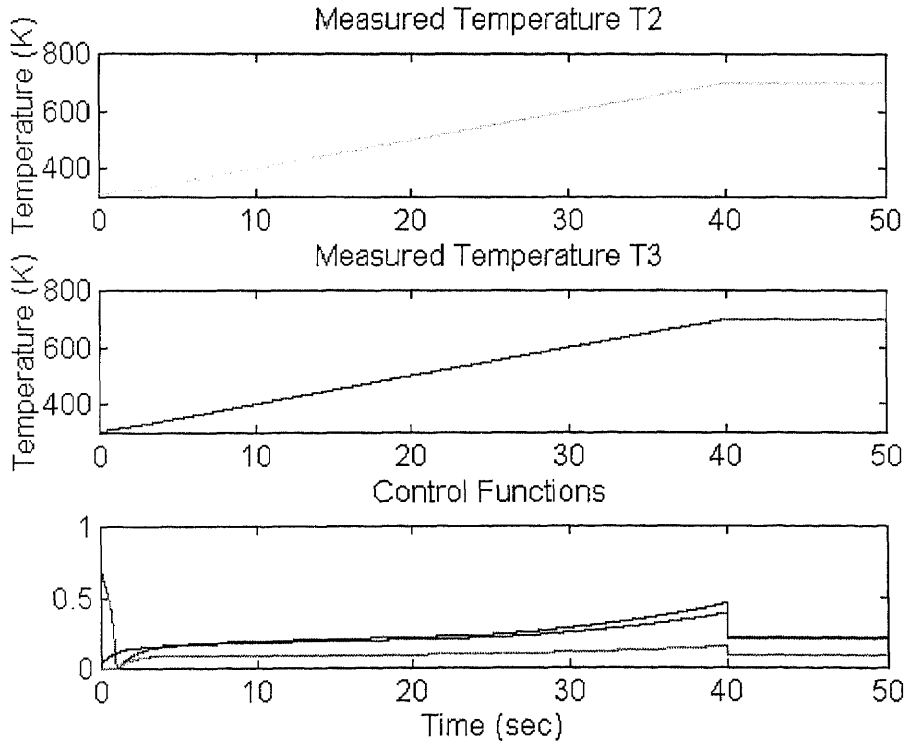


Figure 2.3 Control functions that provide temperature tracking and control

CHAPTER 3

DYNAMIC OBSERVERS FOR WAFER EMISSIVITY ESTIMATION

3.1 Introduction

An observer is a dynamic system [9], usually implemented by a computer model, designed to provide estimates of the states or other parameters of the system that cannot be measured directly. Observers in general are driven by the outputs of the physical (or simulated) system, the plant, and by the inputs to the plant. The plant to control here is an RTP station that consists of the lamps (actuators) and of the wafer whose temperature is the process to be controlled. The inputs to the plant are the control voltages applied to the lamps, while the states (also the outputs) are the measured temperatures at various points of the wafer. The observer is then driven by the measured temperatures from the thermocouples and by the persistent excitation of the temperature trajectory provided by the control voltage applied to the lamps. If the observer is designed correctly, the states of the observer will converge to the states or parameters of the plant, whichever are to be estimated. Using the temperature measurements, we will be able to estimate the parameters of emissivity models and also the states and disturbance parameters. Two popular estimators, a reduced order non-linear estimator, and a State Dependent Riccati Equation Filter were chosen for wavelength dependent emissivity identification and state and parameters estimation.

3.2 Non-linear Reduced Order Observer

The non-linear reduced order observer is based on the method outlined in [10]. The general observer equations for a dynamic process $\dot{x} = f(x,u,\theta)$ are given by:

$$\hat{\theta} = \phi(x) + z \tag{3.1}$$

$$\dot{z} = -\Phi(x) f(x,u,\hat{\theta}) \tag{3.2}$$

where x is the state of the process, u is the control input, and θ is a vector of parameter to be estimated. In addition, $\phi(x)$ is a function appropriately chosen by the designer to guarantee convergence of estimates to actual values and $\Phi(x)$ is its Jacobian matrix.

Given the dynamic process of equation (2.6), repeated here for convenience,

$$\dot{T} = \sum_{j=1}^{n} p_j F_j(T) + 2h^{-1}[S + G(\mathbf{U})] \tag{2.6}$$

where T is the temperature of the wafer at various points, $G(\mathbf{U})$ is the heat flux generated by the vector $\mathbf{U}$ (control voltages applied to the lamp rings) and p_j are the parameters to be estimated, the non-linear reduced order observer can be presented in the form obtained following the above observer algorithm

$$\hat{p}_i = K_i \int F_i(T)dT + z_i, \quad i = 1,\ldots,n \tag{3.3}$$

$$\dot{z}_i = -K_i F_i(T)\left(\sum_{j=1}^{n} \hat{p}_j F_j(T) + 2h^{-1}[S + G(\mathbf{U})]\right) \tag{3.4}$$

where $\hat{p}_i$ are the parameter estimates.

The function $\phi(x) = K_i \int F_i(T)dT$ was chosen to ensure convergence of the estimated parameters to their actual values. The interactive system permits the designer to choose the dynamic model and consequent observer equations for the system under investigation, and test the choice on line. The ultimate objective is again convergence of the estimation error to zero. The requirements for using non-linear reduced order observers for parameter emissivity estimation are that temperature trajectories $T(t)$ be

persistently excited and that the dynamic model equation be linear to parameters to be estimated. The persistent excitation is implemented by the control **U** which should maintain a near uniform temperature trajectory around some specified value. The requirement that the dynamics are linear in the parameters to be estimated implies that $F_j(T)$ in the dynamic equation (2.6) does not contain the parameter p. Equation (2.7) satisfies this requirement. The development of [10] should be consulted to aid in the determination of this type of observer. A block diagram of the architecture of parameter estimation based on reduced order observer is shown in figure 3.1.

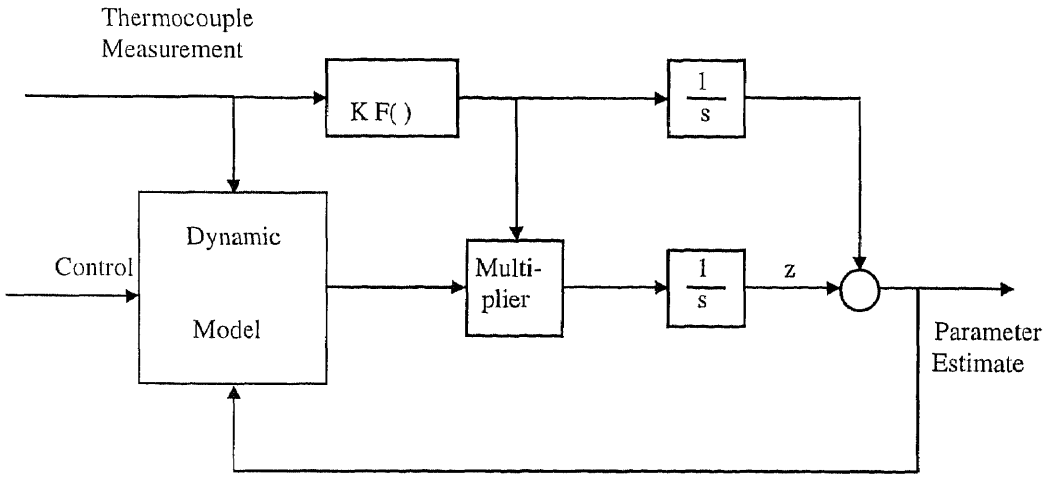


Figure 3.1 Reduced order observer for emissivity estimation

The results of the simulations using the LabVIEW software implementation for this observer using the emissivity model above are shown in chapter 5.

3.3 State Dependent Riccati Equation Filter

In this section, a method referred to as the state dependent algebraic Riccati equation filter (SDREF) is used in the development of states and parameters estimators for use in closed loop control of rapid thermal processing systems. The simulation results of implementing the SDREF as the state and parameter estimator for the adaptive control algorithm outlined in section 2.5 are presented. The SDREF is developed for use with thermocouples as temperature sensors. Therefore, the temperature equations that characterize the development of the RTP model of chapter 2 are used.

The SDREF, derived by constructing the dual of a non linear regulator control design technique [18] which involves the solution of a state dependent Riccati equation, is used to estimate the states $x_n(t)$, and the disturbance parameters $S_n(t)$ as well as the emissivity $\varepsilon(t)$ of the wafer in the RTP system. The state dependent Riccati equation technique and filter design have been developed and proposed in [19]. Consider the stochastic non linear system

$$\begin{aligned} \dot{x} &= f(x,u,t)+w \\ y &= h(x,t)+v \end{aligned} \tag{3.5}$$

where w and v are Gaussian zero-mean white noise processes. The SDREF method is based on a non-unique parameterization that brings the non-linear control system to a linear structure having state dependent coefficients as follows

$$\begin{aligned} \dot{x} &= A(x)x+w \\ y &= C(x)x+v \end{aligned} \tag{3.6}$$

Then, the SDREF is given by the equation

$$\dot{\hat{x}} = A(\hat{x})\hat{x} + K_f(\hat{x})\left[y(x) - C(\hat{x})\hat{x}\right] \tag{3.7}$$

where

$$K_f(\hat{x}) = PC'(\hat{x})R^{-1} \tag{3.8}$$

and P is the positive definite solution to the state dependent Riccati equation

$$A(\hat{x})P + PA'(\hat{x}) - PC'(\hat{x})R^{-1}C(\hat{x})P + Q = 0 \tag{3.9}$$

The matrices $A(x)$ and $C(x)$ were given in section 2.5. The routine that solves the ARE and produces the gain matrix K_f is given in Appendix A. The following simulations for state and parameters estimation were done using a software tool ALSIM that permits solving of differential equations and contains the routine that solves the Riccati equation. The simulations were run using the following values: R = 0.01 and Q = 0.75 diag(3). The gain matrix was updated every 0.1 sec. The results of the simulations when the initial states of the observer were set to the initial states of the system are shown in figure 3.2 for both the real and estimated states. Figure 3.3 illustrates the errors on the states which remain at the same zero order of magnitude throughout the integration interval. Notice, in addition, that, in this case, the filter does not present a convergence transient.

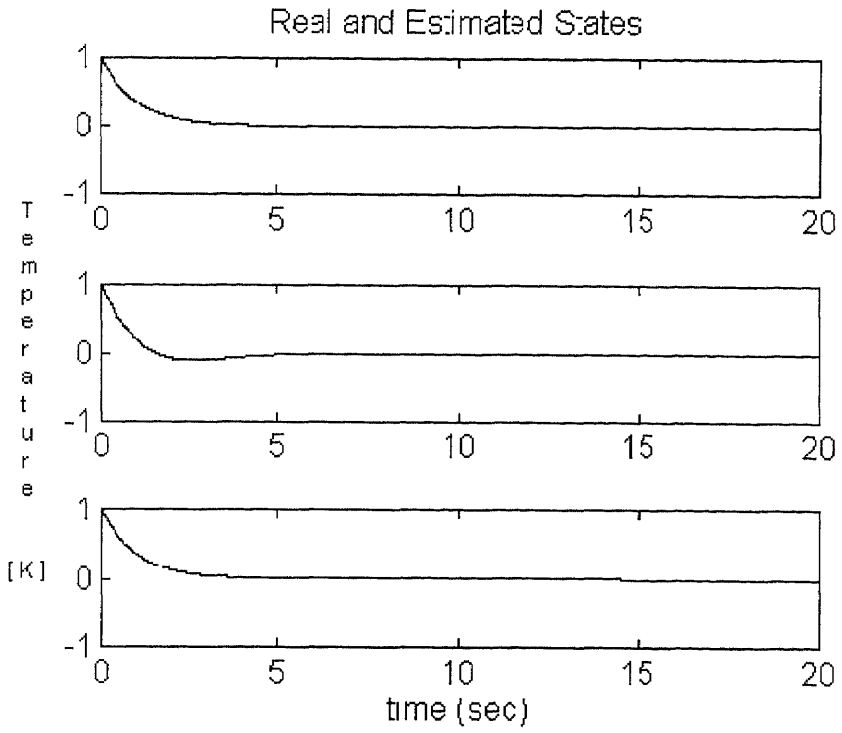


Figure 3.2 Real and estimated states x1,x2, and x3 respectively.

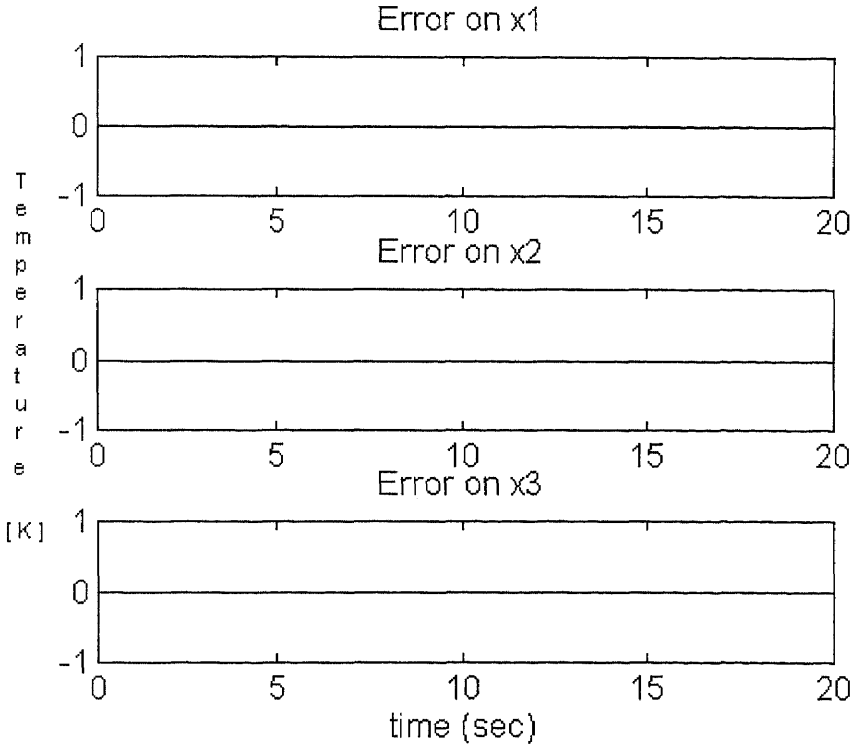


Figure 3.3 Errors on states x1, x2, x3 (same initial conditions)

The results of the simulations when the initial states of the observer were not set equal to the initial states of the system are shown in figures 3.4, 3.5, and 3.6 for both the real and estimated states. Notice that the observer exhibits a smooth behavior. Convergence of the estimated values to the real values is achieved after about 2 seconds.

The unknown parameters of the system, S, are also estimated using the SDREF. These parameters are included in the models augmenting the state matrices.

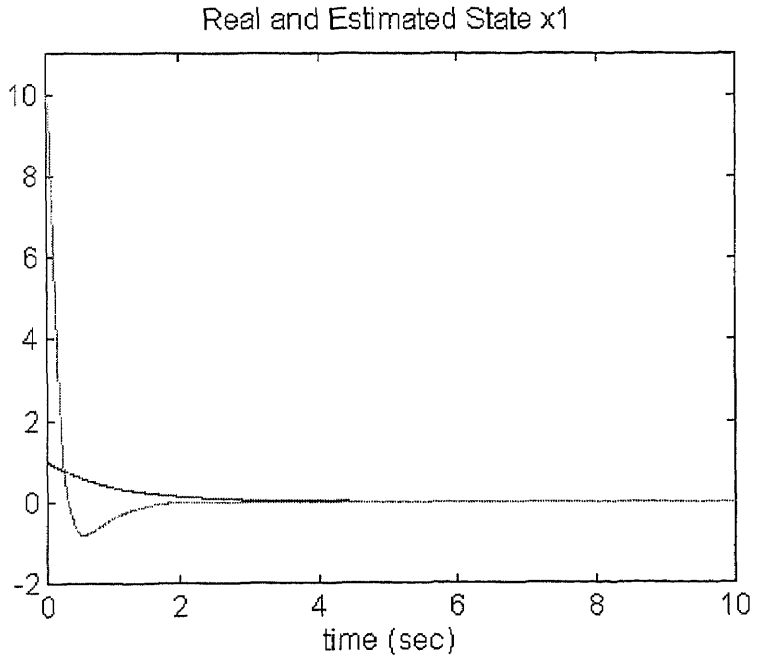


Figure 3.4 Real and estimated state x1

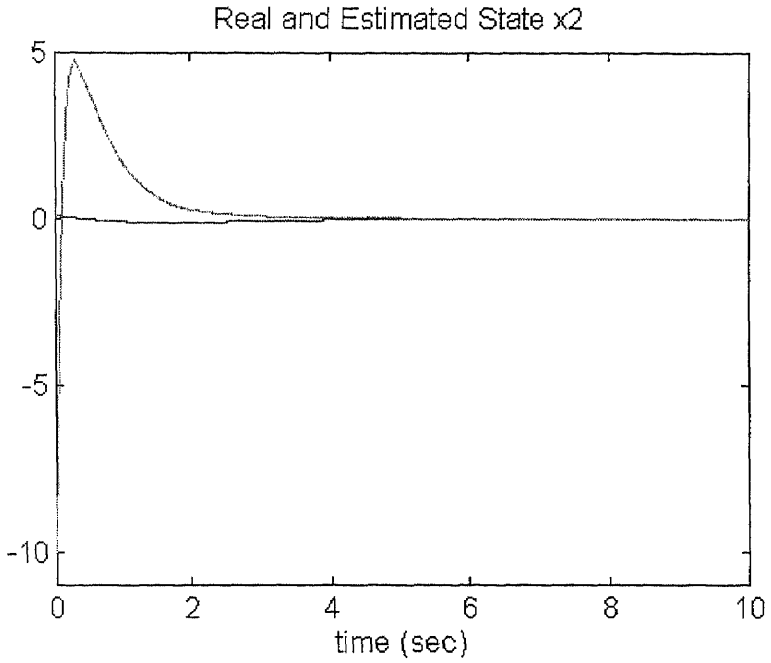


Figure 3.5 Real and estimated state x2

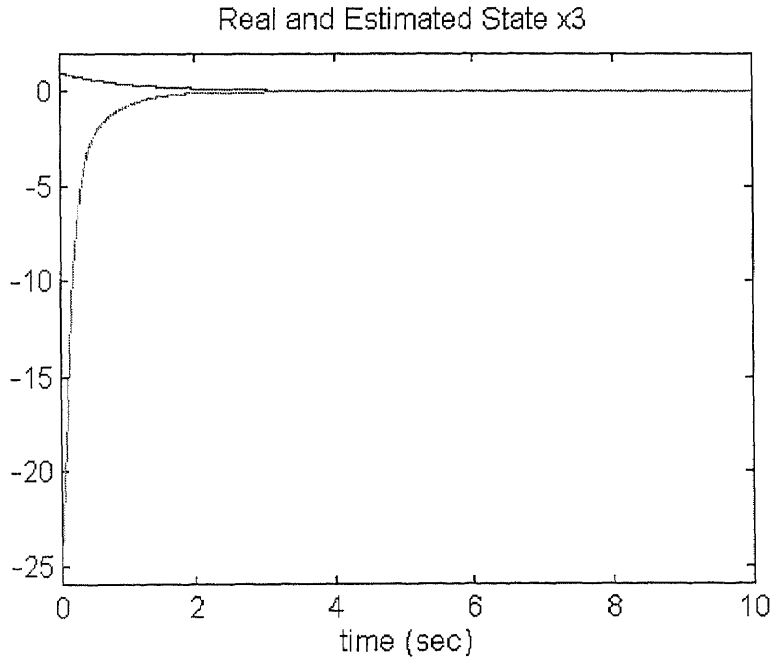


Figure 3.6 Real and estimated state x3

The estimated unknown parameters S are shown in figure 3.7. The results of the closed loop control simulation in which the estimated parameters were fed back into the controller are shown in figure 3.8.

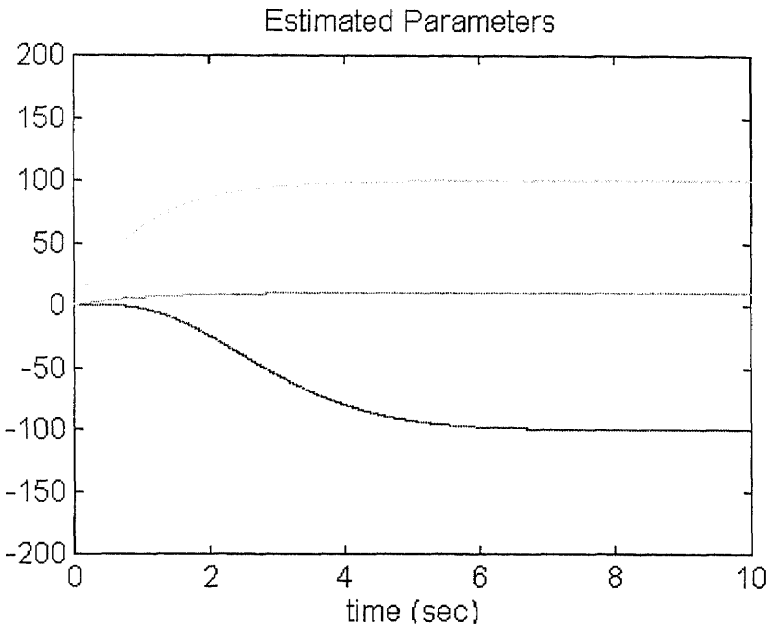


Figure 3.7 Estimated Parameters S_1=100, S_2= -100, S_3=10

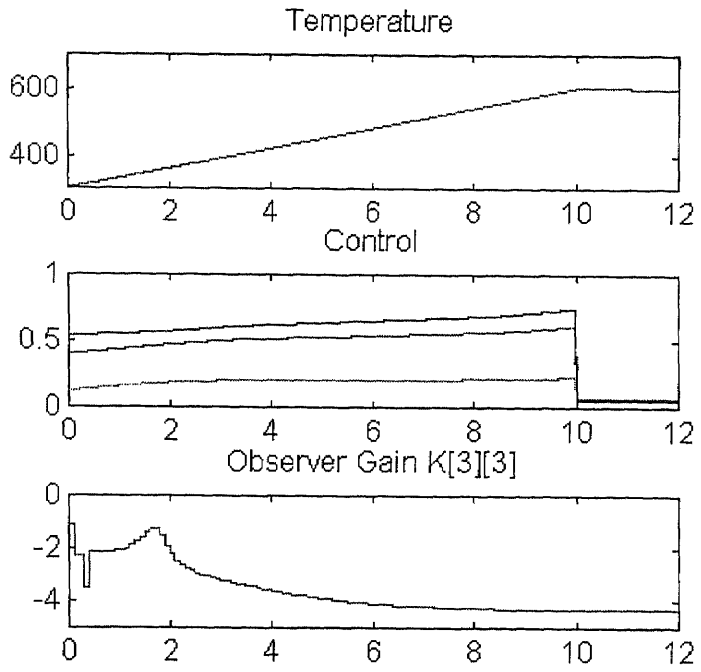


Figure 3.8 Closed-loop control with SDREF as parameter estimator

Figure 3.9 shows plots comparing the measured temperatures against the reference trajectory as well as comparison of two measured temperatures against one another. The plots show that the convergence of the measured temperatures to the reference temperature is fast even during the ramp up period. The uniformity of the measured temperatures is observed to be quite good.

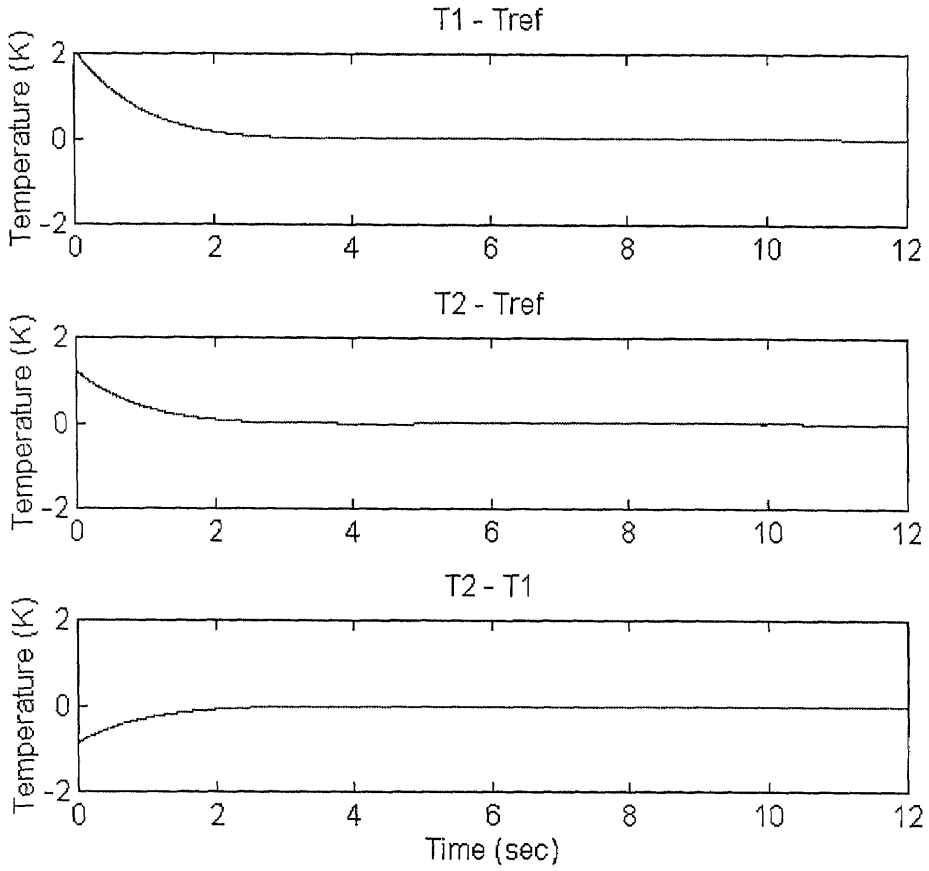


Figure 3.9 Comparison of measured temperature vs. reference trajectory

3.4 SDREF for Emissivity Estimation

This observer has better application when the model chosen is a non linear function of parameters. In fact this model is applied to the equation of dynamics (2.8) for the emissivity model (2.1). This model, which utilizes thermocouple sensors for temperature measurements, suggests an equation in a (non-unique) state-dependent matrix form called "parametrization":

$$\begin{pmatrix} \dot{T} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} F_{TT}(\mathbf{p},T) & \mathbf{F}_{\mathrm{Tp}}(\mathbf{p},T) \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} T \\ \mathbf{p} \end{pmatrix} + \begin{pmatrix} 2h^{-1}[S+G(\mathbf{U}) \\ \mathbf{0} \end{pmatrix} + \mathbf{w} \qquad (3.10)$$
$$y = T + v$$

where $\mathbf{w}$ is an artificially added random vector (white noise) with covariance matrix $\mathbf{W}$, and v is an artificially added random scalar (white noise) with covariance V. Covariance matrices $\mathbf{W}$ and V can be chosen to optimize the behavior of the filter. The prudent design of the SDREF is a procedure that provides an intelligent choice of parametrization, i.e. matrices F_{TT} and $\mathbf{F}_{Tp}$, and the noise covariance matrices $\mathbf{W}$ and V. The implementation of the SDREF consists of an application of Kalman filtering, which is a solution of Riccati equation for state-dependent matrices of equation (3.5).

SDREF, as well as the non-linear reduced order observer, requires a persistently excited temperature trajectory. The necessity of the persistent excitation is illustrated in figure 3.10, where SDREF was used for estimation of total effective emissivity of the wafer degraded by noise. Convergence of the estimate to the actual value of emissivity is reliable when the temperature is changing quickly. However, the estimate diverges when the temperature approaches steady state value. Persistent excitation prevents the divergence, therefore it is of paramount importance. Figure 3.10 demonstrates the necessity of persistent excitation. The emissivity estimate, with the initial value of 0.7, converges to the noisy actual emissivity before the temperature approaches a steady state value at the time of about 12 seconds. At this time it is shown that the emissivity's estimate starts to diverge. Persistent excitation is necessary at this moment to guarantee an accurate estimation.

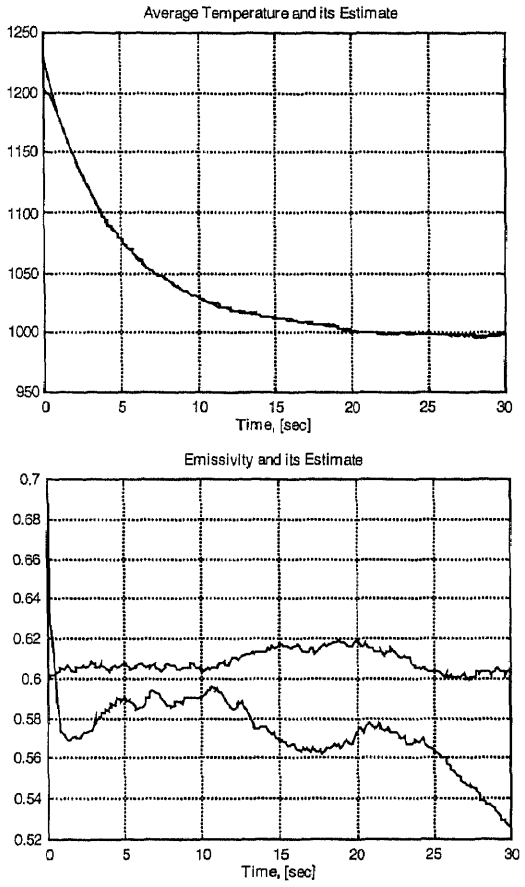


Figure 3.10 Temperature profile, emissivity, and their estimates

CHAPTER 4

IMPLEMENTATION OF THE SYSTEM

4.1 Introduction

The interactive system has been implemented on a pentium based PC using LabVIEW® [11] and G Math Toolkit [12]. The advantage of using this software is a new approach for solving analytical problems. The use of graphical programming G simplifies the task of programming long lines of written code. The modularity of the program can be maintained with the use of virtual instruments (VI) and subVI that can be programmed independently, called and linked together by the main program. The power and speed of the pentium processor is enough for the purpose of implementing the control algorithm and the estimation algorithm for application in an RTP station.

4.2 LabVIEW Features

LabVIEW® is a graphical programming tool developed by National Instruments with which one can graphically illustrate a real-time system as a block diagram. It is particularly adept for applications where real-time measurements and control of variables are needed. Data is collected with appropriate data acquisition boards, it is analyzed with flexible and powerful tools and it is graphically presented in a user-friendly environment. The virtual instrument (VI) is responsible for data acquisition and control, data analysis, and data presentation. VIs are the software modules each containing some useful part of the program that can be run independently. An icon can be assigned to each VI so that it can be called during the compiling of the program to perform the specific function for which it was designed. A VI can also be reused at a later time for other applications, even

if it was not designed specifically for them. In addition, another user has the freedom to modify the virtual instrument to meet his changing application needs. For example, a VI can be built that uses some data analysis algorithm to transform acquired or simulated data from another VI into useful information; after being analyzed, the same data can be passed to another VI that acts like a graphical user interface (GUI) to display it in a friendly way. A standard architecture of the example above for measurement and instrumentation is shown in figure 4.1.

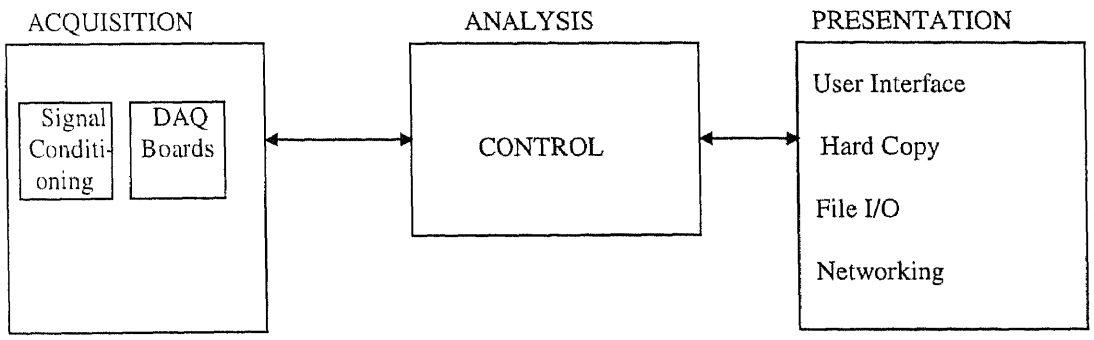


Figure 4.1 A standard architecture for measurement and instrumentation

The acquisition component is in general implemented by a combination of hardware and software, unless data is created by simulation, in which case only software is used to generate data. The analysis component and presentation component are implemented by software. The data flow is shown in the block diagram. With graphical programming, which is the fundamental characteristics of LabVIEW, it is easy to control the data flow.

LabVIEW features a graphical compiler that generates optimized compiled code. VIs execute at speeds comparable to those of compiled programs. LabVIEW version 4.0

running on a 120 MHz Pentium processor based PC using Windows 95® operating system was used for the work in this thesis.

4.3 Programming in LabVIEW

The concepts of object oriented and data-flow programming are the basis of LabVIEW modern programming approach. LabVIEW uses a block diagram programming methodology. The block diagram approach to programming is consistent with the design notation of flow charts and block diagrams. The development of automated measurements and instrumentation systems is simplified by this approach to programming. Instead of using lengthy codes of conventional, text based languages, block diagrams are used with several advantages. First of all, block diagrams are easily interpreted. Second, this methodology is inherently self-documenting. With front panels, block diagrams, and icons, a complete pictorial description of the system is obtained. Examples of front panels, block diagrams, and icons for our implementation are shown in figures 4.2, and 4.3.

The front panels (figure 4.2) provide a graphical user interface and intuitive control of instrumentation. Front panels introduce a new level of interactivity to real-time measurements and analysis. In the interactive system for emissivity estimation, we use the core feature of the G Math Tool which makes manipulation of formulae on LabVIEW front panels possible.

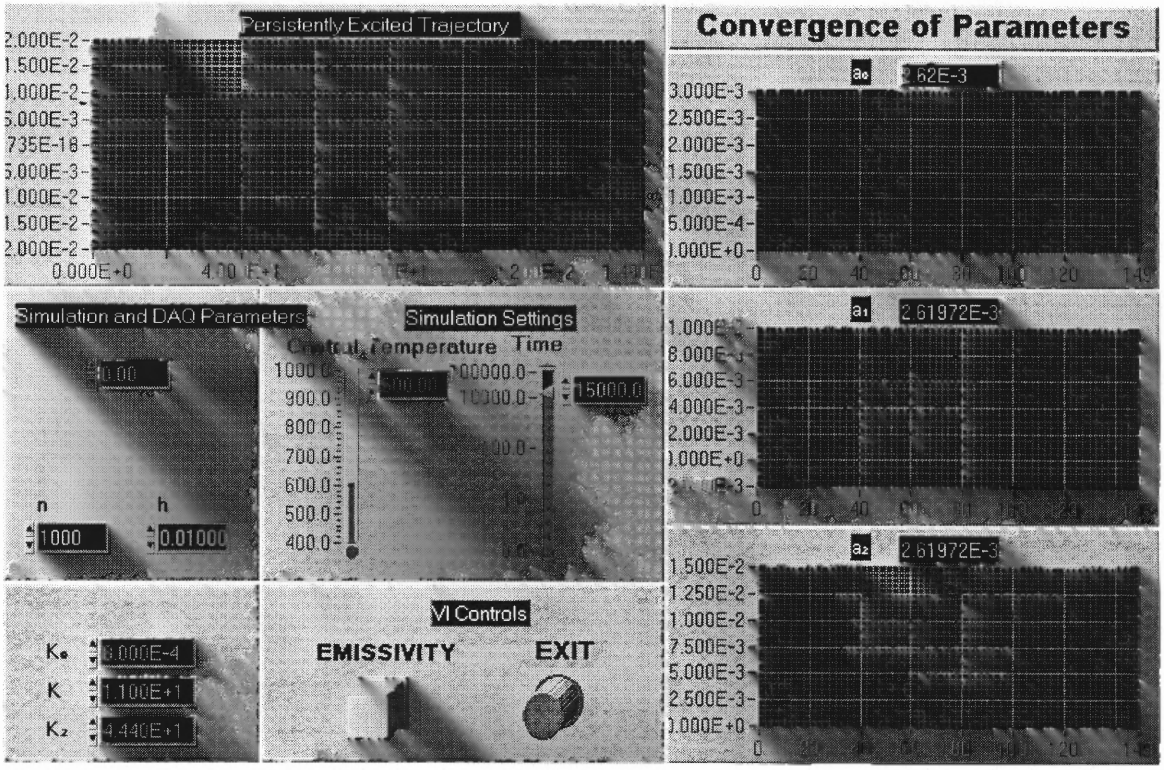


Figure 4.2 LabVIEW front panel interface

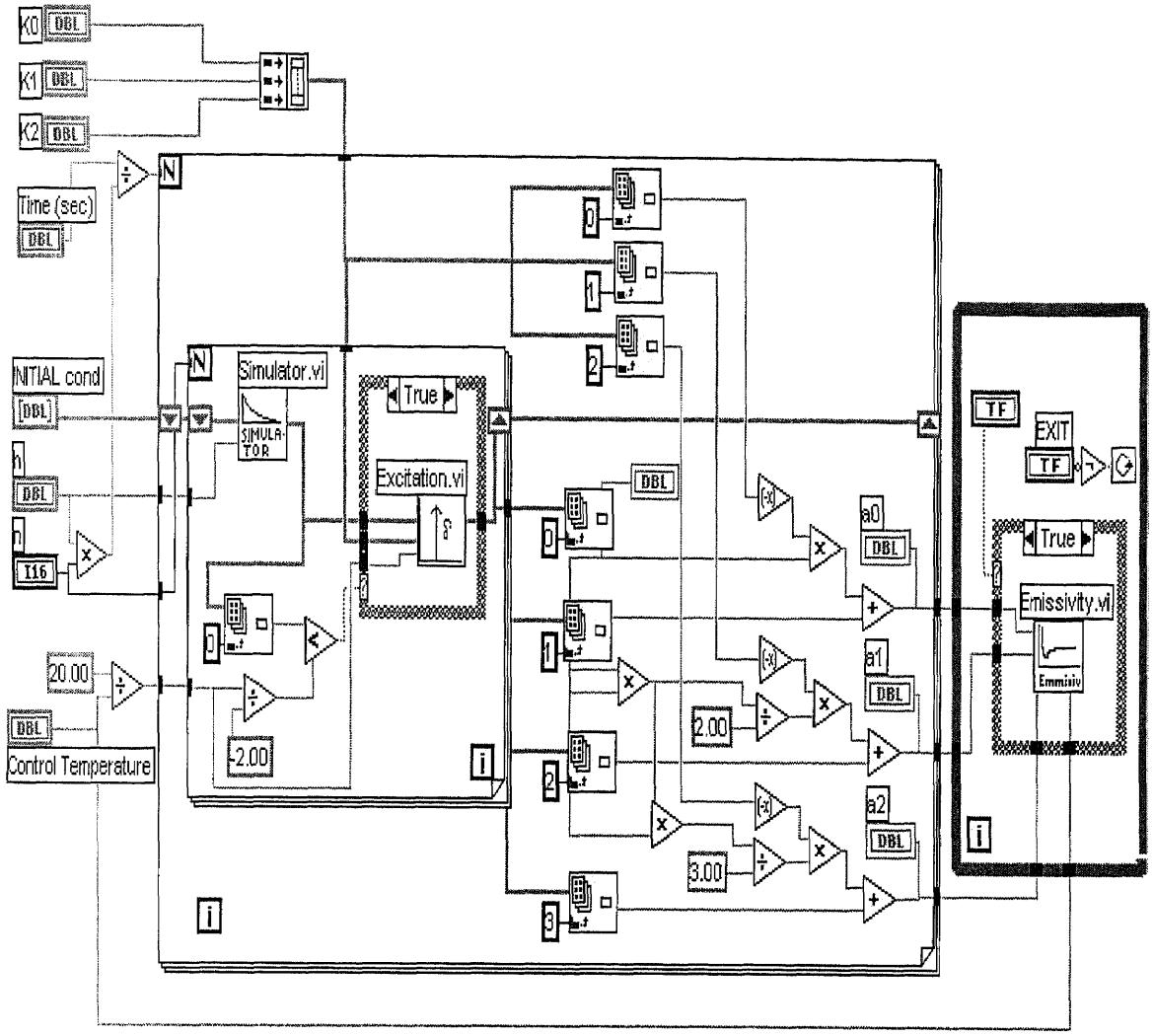


Figure 4.3 LabVIEW block diagram for process analysis

The block diagram (figure 4.3) is where the actual programming is implemented. This type of programming is called data-flow programming. Data-flow programming dictates that an object may not execute until all of its inputs are available, and the object's outputs are not available until the object's function is completed. Thus, the flow of data between connected objects controls the execution order. The execution order is not constrained to the sequential order of lines from a text-based program. Once the data is available, it follows the path designed by simply connecting functional blocks together. Even multiple

data paths, and consequently simultaneous operations can be performed. As in traditional, text-based programming languages, the need of structures that control the block diagram function execution is of primary importance. Program control structures included in LabVIEW are Sequence, If Statement, For Loop, and While Loop. These structures are graphically depicted as border structures like those shown in figure 4.3. Just as a code is imbedded into the lines of a structure in a conventional programming language, graphical functions and icons are placed within the borders of graphical structures.

Modularity of the program is guaranteed by this approach to software design. In fact, LabVIEW is a hierarchical system. Any virtual instrument can serve as a sub-routine for another VI. A VI can be used as a sub-VI by adding its icon in the block diagram (figure 4.3). A sub-VI can be opened and run interactively to verify correct operation. If used as a sub-VI, the data passed to the icon is the same information supplied through the VI's front panel. To the application of the work in this thesis, the interactivity results in a programming environment in which we can rapidly design, combine, interchange, and share VIs.

4.4 System Architecture

A block diagram of the system is presented in figure 4.4. Every block inside the dashed line is implemented by the PC using codes in LabVIEW. The RTP simulator creates the temperature of the wafer as it would be measured by the thermocouple sensors in accordance with the assumed model. However, the current implementation can be easily upgraded to incorporate optical pyrometers for temperature measurement. The measured temperature is fed into the controller algorithm whose purpose is to apply the persistent

excitation to the lamps necessary to guarantee an accurate estimation. The necessity for persistent estimation was explained in the observer design chapter and a graphical explanation will be given in the next chapter where the simulation results will be shown.

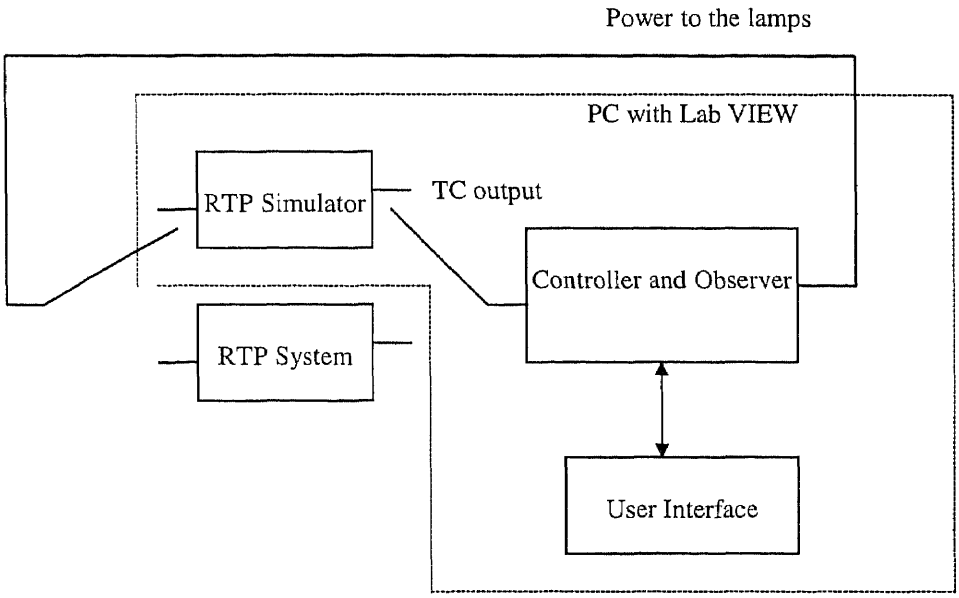


Figure 4.4 Architecture of the interactive system of emissivity estimation

The temperature measurement is also fed into the observer which is responsible for the estimation of emissivity. The user interface performs the monitoring of the simulation experiments and it is the interactive environment. The user interface block includes the virtual instruments responsible for the user interaction with the system. Controllers and observers parameters and formulas can be changed prior to the simulation experiment to incorporate different models of wafer emissivity and RTP heat transfer dynamic models depending on the application. The different models can be selected interactively on site depending on the experiment. The front panel of figure 4.5 shows that by a point and click operation a model of the wavelength dependent emissivity can be chosen according

to the experiment being conducted. If, for example, an emissivity model which is linear with respect to parameters is chosen for the experiment, then the box that performs an estimation of the emissivity of the wafer for a linear model must be pointed and clicked. In the LabVIEW environment this corresponds to model 1 because when the mouse is brought close to the model 1 box an help window appears that describes which emissivity model is used if this box is clicked. When the box model 1 is clicked the following front panel appears as in figure 4.6 and the simulation starts with the estimation of the parameters necessary for the estimation of emissivity.

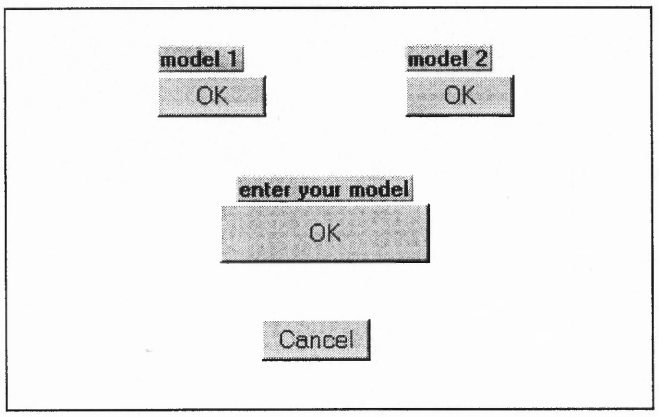


Figure 4.5 Front panel for the selection of the model

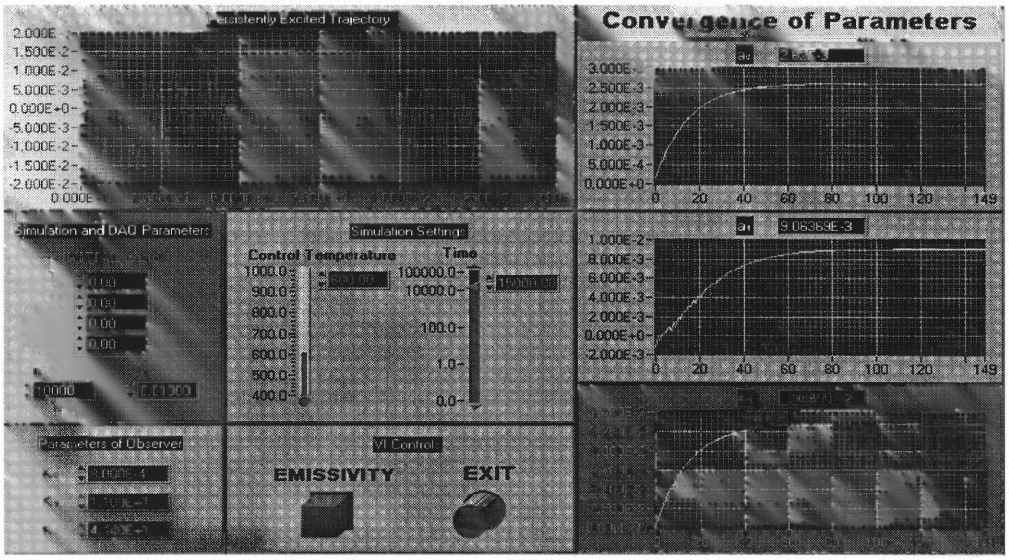


Figure 4.6 Front panel for the estimation of parameters

CHAPTER 5

SIMULATION RESULTS

5.1 Introduction

The estimation of wavelength-dependent emissivity of the wafer using the simulated interactive system is presented here for the conditions in which the emissivity model is a linear function of parameters. As a reference, we take previous experimental runs done on the same model [8] but using our different implementation method which uses LabVIEW programming. The results of [8] were satisfactory so we use the same approach of solving the problem of estimating the emissivity of the wafer in an RTP chamber. The purpose here is to verify that the implementation using LabVIEW programming works correctly and gives comparable results. The theoretical work, presented in [7], is summarized here for reference purpose only.

5.2 Theory

In chapter 2, generic equations that identify different emissivity models were given and observers for the different models were proposed in chapter 3. To perform the simulation of estimation of emissivity a specific model was selected after investigations with Fourier Transform Infrared Spectrometry [7] have shown that a valid approximation of the wafer emissivity ε as a function of wavelength λ and temperature T is

$$\varepsilon(\lambda, T) = p_0 + p_1\lambda^{-1} + p_2\lambda^{-2} \tag{5.1}$$

The emissivity $\varepsilon(\lambda, T_0)$ can be estimated at the given temperature T_0 by estimating the temperature dependent parameters $p_l(T)$. The equation for the emission function $E(T)$, the

heat flux emitted via radiation from the unit area of the wafer with above emissivity is [13]

$$E(T) = \alpha_0 p_0 T^4 + \alpha_1 p_1 T^5 + \alpha_2 p_2 T^6 \tag{5.2}$$

where $\alpha_0 = \sigma = 5.67 \cdot 10^{-8}$ [W /m^{2}K^4] is the Stefan-Boltzmann's constant, $\alpha_1 = 1.51 \cdot 10^{-5}$ [W/m^{3}K^5], and $\alpha_2 = 5.15 \cdot 10^{-3}$[W / m^{4}K^6] [13]. As explained in the chapter on observer design, a uniform and persistently excited temperature trajectory near T_0 is needed in order to estimate the parameters p_l , l=0,1,2. A TI designed RTP equipment and an adaptive closed-loop temperature control have been used to generate the required persistent excitation. The generated reference temperature trajectory can be described by the following equations:

$$T(t) = T_0 x(t),\ x(t) = 1 + \Delta x(t),\ |\Delta x| \approx 10^{-2} \tag{5.3}$$

where $x(t)$ is calculated from the following differential equation:

$$\dot{x} = \theta_0 - \theta_4 x^4 - \theta_5 x^5 - \theta_6 x^6 + g(\mathbf{U}). \tag{5.4}$$

In the above equation, $g(\mathbf{U})=2G(\mathbf{U}) / (\rho c(T_0)h)$, $\theta_0=2S / (\rho c(T_0)T_0 h)$, and θ_{4+l} is related with p_l by the following equation:

$$p_l = \frac{\rho c(T_0) h}{2\alpha_l T_0^{3+l}} \theta_{4+l},\ l = 0,1,2 \tag{5.5}$$

The differential equation (5.4) contains parameters that are not easy to identify because for very small $\Delta x(t)$ it is difficult to distinguish higher powers of $x(t)=1+\Delta x(t)$. Therefore, the following approximation is used

$$(1+\Delta x)^n = 1 + n\Delta x + \frac{n(n+1)}{2}\Delta x^2 + O(10^{-6}) \tag{5.6}$$

and the equation for $\Delta x(t)$ can be expressed as follows:

$$\dot{\Delta x} = -a_0 - a_1 \Delta x - a_2 \Delta x^2 + g(\mathbf{U}) \tag{5.7}$$

where parameters a_l are related to θ by the following equations:

$$\begin{aligned} \theta_4 &= 15a_0 - 5a_1 + a_2 + 15\theta_0 \\ \theta_5 &= -24a_0 + 9a_1 - 2a_2 - 24\theta_0 \\ \theta_6 &= 10a_0 - 4a_1 + a_2 + 10\theta_0 \end{aligned} \tag{5.8}$$

By estimating parameters a_l , the emissivity of the wafer can be estimated using the equations above. The parameter θ_0 can be assumed to be zero if vacuum condition is maintained in the chamber and its walls are cool as it is the case in the experimental RTP system. In other situations, parameter θ_0 should be estimated.

5.3 Dynamic Observer

The equations of the dynamic observer chosen for this particular model, and used to estimate parameters a_l of equation (5.8) are shown below:

$$\begin{aligned} \hat{a}_l &= -K_l \frac{\Delta x^{l+1}}{l+1} + z_l, \quad l = 0,1,2 \\ \dot{z}_l &= K_l \Delta x^l (-\hat{a}_0 - \hat{a}_1 \Delta x - \hat{a}_2 \Delta x^2 + g(\mathbf{U})) \end{aligned} \tag{5.9}$$

where $\hat{a}_l$ are the parameters estimates, $K_l > 0$ are the gains, and z_l are the internal states of the observer.

The equations of states z_l of the observer can be derived after the control function $g(\mathbf{U})$ is adequately chosen. The purpose is to remove the control function from the observer state equations. An impulse function was chosen as the control function[7].

Using the impulse function mathematical definition $g(\mathbf{U})=\int_{t}^{t+\tau}\delta(\tau)d\tau=(A/\Delta)\tau$, the state Δx can be derived integrating equation (5.7) at time $t+\tau$ as

$$\Delta x(t+\tau)=\Delta x(t-0)-\int_{t}^{t+\tau}a_0 d\tau-a_1\int_{t}^{t+\tau}\Delta x(\tau)d\tau-a_2\int_{t}^{t+\tau}\Delta x(\tau)^2 d\tau+\frac{A}{\Delta}\tau \qquad (5.10)$$

for which the result is obtained

$$\Delta x(t+\tau)=\Delta x(t-0)-a_0\tau-a_1\overline{\Delta x}\tau-a_2\overline{\Delta x}^2\tau+(A/\Delta)\tau \qquad (5.11)$$

where $\Delta x(t\text{-}0)$ are the initial conditions and Δx bar and Δx^2 bar are average approximations of integrals. The state equations for the observer are obtained by integrating equations (5.9), expanded as follow for convenience:

$$\begin{aligned}
\dot{z}_0 &= K_0\left\{(K_0\Delta x+z_0)+\left(K_1\frac{\Delta x^2}{2}+z_1\right)\Delta x+\left(K_2\frac{\Delta x^3}{3}+z_2\right)\Delta x^2+g(\mathbf{U})\right\}\\
\dot{z}_1 &= K_1\Delta x\left\{(K_0\Delta x+z_0)+\left(K_1\frac{\Delta x^2}{2}+z_1\right)\Delta x+\left(K_2\frac{\Delta x^3}{3}+z_2\right)\Delta x^2+g(\mathbf{U})\right\} \qquad (5.12)\\
\dot{z}_2 &= K_2\Delta x^2\left\{(K_0\Delta x+z_0)+\left(K_1\frac{\Delta x^2}{2}+z_1\right)\Delta x+\left(K_2\frac{\Delta x^3}{3}+z_2\right)\Delta x^2+g(\mathbf{U})\right\}
\end{aligned}$$

As $\tau\rightarrow 0$, for impulse function approximation, the following state equation and observer state equations are obtained:

$$\Delta x(t+0)=\Delta x(t-0)+A \qquad (5.13)$$

$$\begin{aligned}
z_0(t+0) &= z_0(t-0)+K_0A\\
z_1(t+0) &= z_1(t-0)+K_1\left(\Delta x(t-0)A+A^2/2\right) \qquad (5.14)\\
z_2(t+0) &= z_2(t-0)+K_2\left((\Delta x(t-0))^2A+\Delta x(t-0)A^2+A^3/3\right)
\end{aligned}$$

where equation 5.13 is derived from equation 5.11 noticing that the expression $(A/\Delta)\tau=A$ as $\tau\rightarrow 0$ and where $z_i(t\text{-}0)$, $i=0,1,2$ are the initial observer states,

$A = (T(t+0) + T(t-0))/T_0$ is the amplitude of the δ-impulse function, t is the time at which the impulse function is applied, $T(t\text{-}0)$ is the temperature reading just before the impulse function is applied, and $T(t\text{+}0)$ is the temperature reading just after the impulse function is applied. The above observer state equations are obtained noticing that when integrating equations 5.12, the following integral's values approach zero as $\tau \to 0$

$$\begin{matrix} \int_t^\tau K_0 \Delta x(t+\tau) + z_0 d\tau \\ \int_t^\tau K_1 \frac{\Delta x^2(t+\tau)}{2} + z_1 d\tau \qquad \to 0 \text{ as } \tau \to 0 \\ \int_t^\tau K_2 \frac{\Delta x^3(t+\tau)}{3} + z_2 d\tau \end{matrix} \tag{5.15}$$

5.4 Simulation Study

Experimentally, we want the impulse to change the temperature from T_0-10°K to T_0+10°K. This control cannot be implemented directly on the experimental setup but the desired temperature trajectory can be obtained using the following experiment. The control is turned off when the wafer reaches a temperature T_0+100°K and the wafer is let to cool down to a temperature T_0-100°K at which the closed-loop adaptive temperature control is turned on to heat the wafer up to the temperature T_0+100°K and the cycle is repeated this way. The part of the trajectory in the cooling down period, to be specific, from T_0+10°K to T_0-10°K was recorded on file in previous experiments using the current real time adaptive controller implementation[8]. The data recorded on file represents the trajectory equivalent to the temperature output as if the impulse control was applied[8].

The simulation run using LabVIEW program shows convergence of the estimated parameters a_i to the values calculated using an initial wavelength-dependent emissivity formula

$$\varepsilon(\lambda) = 0.3076 - 1.653 \cdot 10^{-6} \lambda^{-1} + 2.815 \cdot 10^{-12} \lambda^{-2} \qquad (5.16)$$

at temperature T_0 = 600°K. The coefficients derived from above equations are $a_0 \approx 2.62 \cdot 10^{-3}$, $a_1 \approx 8.93 \cdot 10^{-3}$, and $a_2 \approx 1.13 \cdot 10^{-2}$. The convergence of estimated parameters to the above values is shown in figure 5.1. Figure 5.2 shows the emissivity function calculated from the estimated parameters.

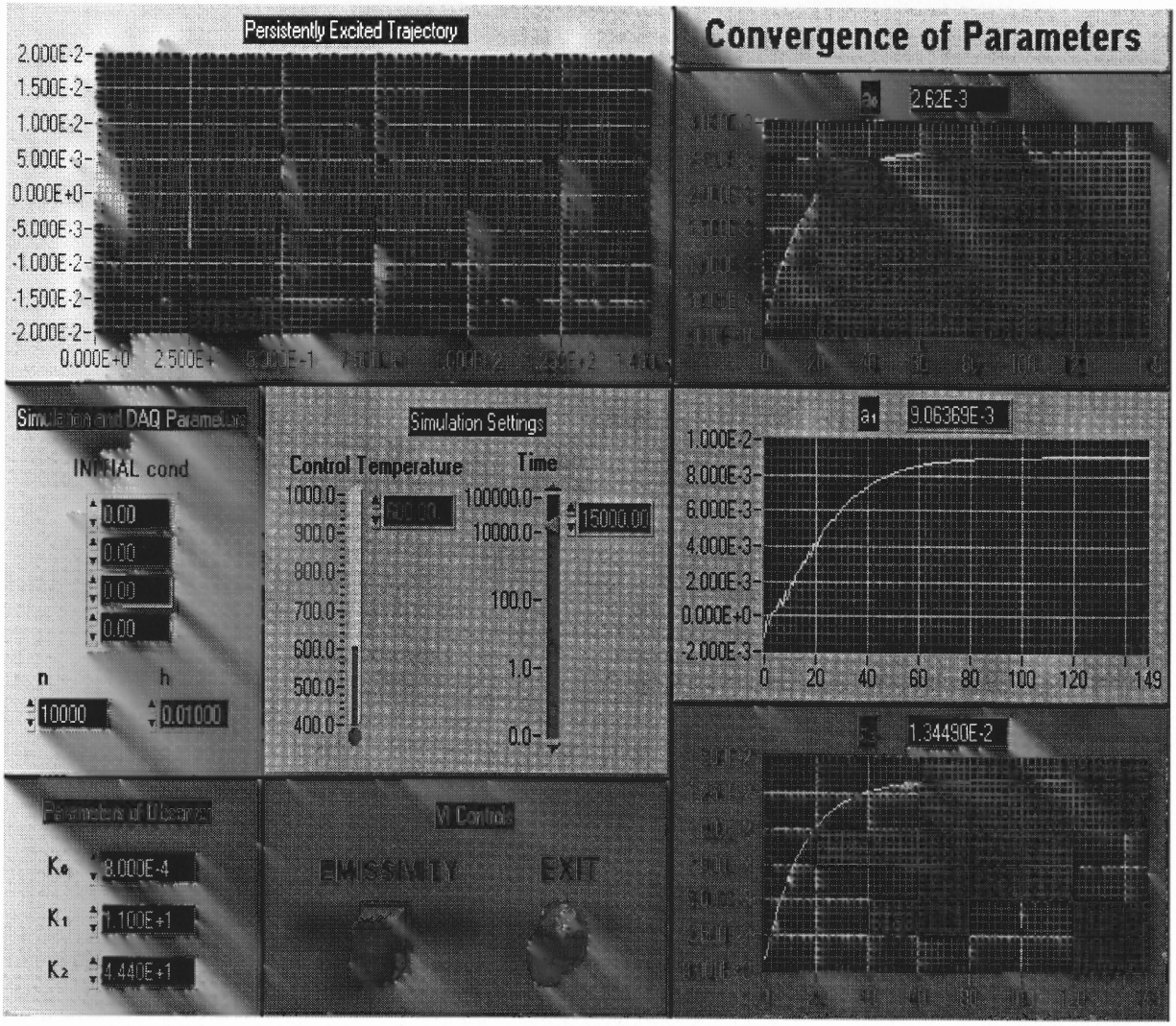


Figure 5.1 Estimation of parameters in simulation

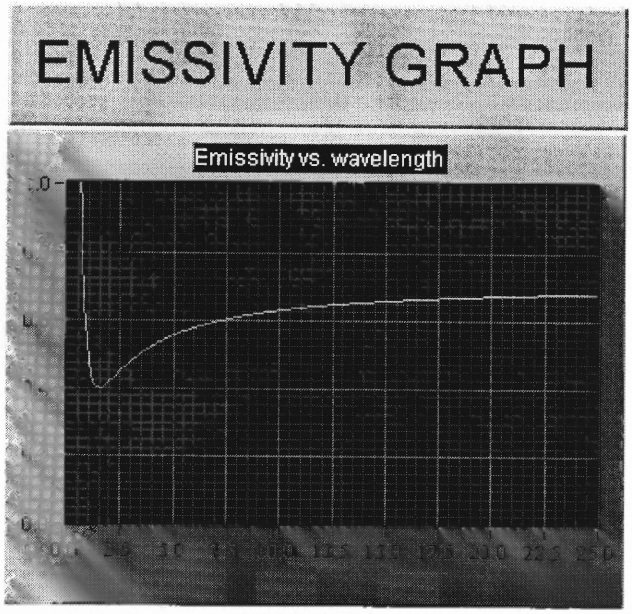


Figure 5.2 Emissivity function of estimated parameters

CHAPTER 6

CONCLUSION

This thesis has attacked the challenging problem of estimating unknown parameters, among which the silicon wafer's emissivity as a function of wavelength emitted, when the wafer is being processed in a Rapid Thermal Processing system. By building an interactive LabVIEW based application we have provided a solution that utilizes a physical model and an appropriate observer to estimate the emissivity. The importance of the interactive interface cannot be overemphasized. This involves measuring the temperature of the wafer being processed by means of pyrometers and display results in real time including data analysis. In addition, initial conditions and other parameters must be easily modifiable during experiments. The system was designed to provide enough flexibility during the real time experiment.

As a way of improving the performance of the current control system, the simulation of the system using the State Dependent Riccati Equation Filter was implemented and the convergence of the observer was demonstrated. The simulation was carried out with the observer inserted in the closed-loop control system were the controller was already designed using an adaptive control algorithm. The proposed observer provided good estimation of the states. In addition, using the SDRE filter we were able to estimate the unmodelled heat flux disturbance parameters. The latter parameters were assumed to be constant so that parametrization of the system in an augmented form was possible. The SDRE method provides a systematic approach for the design of nonlinear observers.

APPENDIX A

RTP ALSIM SIMULATION PROGRAM LISTING

RTP.H

```
/*    This function interpolates from a table                    */
/*    to find the specific heat as a function                    */
/*    of Temperature                                             */

double c_heat(double T)
{
     double ret;
     int i;
     double Ttab[9]={
     0.,100.,200.,400.,600.,800.,1000.,1200.,3000.
     };
     double ctab[9]={
     256.,256.,549.,780.,856.,900.,934.,955.,955.
     };


     for(i=1;T>Ttab[i];i++);
     ret=ctab[i-1]+(ctab[i]-ctab[i-1])*(T-Ttab[i-1])
       /(Ttab[i]-Ttab[i-1]);
     return(ret);
}



 /*   This function interpolates from a table to find the heat capacity  */
 /*    k, as a function of temperature                                    */


double k(double T)
{
     double Ttab[9]={
     0.,100.,200.,400.,600.,800.,100.,1200.,3000.
     };
     double ktab[9]={
     884.,884.,264.,98.9,61.3,42.2,31.2,25.7,25.7
     };
     int i;
     double ret;

     for(i=1;T>Ttab[i];i++);
     ret=ktab[i-1]+(ktab[i]-ktab[i-1])*(T-Ttab[i-1])
       /(Ttab[i]-Ttab[i-1]);
     return(ret);
}



 /*   This function returns the expression (2*emissivity*sigma/h) as a  */
 /*   function of Temperature                                            */
```

```
double E(double T)
{
      double Ttab[5]={
      0.,550.,700.,740.,3000.
      };
      double etab[5]={
      0.2,0.2,0.5,0.7,0.7
      };
      int i;
      double e;

      for(i=1;T>Ttab[i];i++);
      e=etab[i-1]+(etab[i]-etab[i-1])*(T-Ttab[i-1])
       /(Ttab[i]-Ttab[i-1]);
      return(2.*e*Sigma/h);
}

/**********          Voltages for the lamp rings          **************/

double volt1(double u)
{
      double utab[6]={
      0.,0.19,0.31,0.39,0.73,1.
      };
      double vtab[6]={
      0.,1.,2.,3.,4.,5.
      };
      int i;
      double ret;

      for (i=1;u>utab[i];i++);
      ret=vtab[i-1]+(vtab[i]-vtab[i-1])*(u-utab[i-1])
        /(utab[i]-utab[i-1]);
      return(ret);
}

double volt2(double u)
{
      double utab[2]={
      0.,1.
      };
      double vtab[2]={
      0.,1.
      };
      int i;
      double ret;

      for (i=1;u>utab[i];i++);
      ret=vtab[i-1]+(vtab[i]-vtab[i-1])*(u-utab[i-1])
        /(utab[i]-utab[i-1]);
      return(ret);
}

double volt3(double u)
{
```

```
        double utab[6]={
        0.,0.046,0.29,0.53,0.8,1.
        };
        double vtab[6]={
        0.,1.,2.,3.,4.,5.
        };
        int i;
        double ret;

        for (i=1;u>utab[i];i++);
        ret=vtab[i-1]+(vtab[i]-vtab[i-1])*(u-utab[i-1])
          /(utab[i]-utab[i-1]);
        return(ret);
}

double g1(double u)
{
        double gtab[6]={
        0.,0.19,0.31,0.39,0.73,1.
        };
        double vtab[6]={
        0.,1.,2.,3.,4.,5.
        };
        int i;
        double ret;

        for (i=1;u>vtab[i];i++);
        ret=gtab[i-1]+(gtab[i]-gtab[i-1])*(u-vtab[i-1])
          /(vtab[i]-vtab[i-1]);
        return(ret);
}

double g2(double u)
{
        double gtab[2]={
        0.,1.
        };
        double vtab[2]={
        0.,1.
        };
        int i;
        double ret;

        for (i=1;u>vtab[i];i++);
        ret=gtab[i-1]+(gtab[i]-gtab[i-1])*(u-vtab[i-1])
          /(vtab[i]-vtab[i-1]);
        return(ret);
}

double g3(double u)
{
        double gtab[6]={
        0.,0.046,0.29,0.53,0.8,1.
        };
        double vtab[6]={
```

```
    0.,1.,2.,3.,4.,5.
    };
    int i;
    double ret;

    for (i=1;u>vtab[i];i++);
    ret=gtab[i-1]+(gtab[i]-gtab[i-1])*(u-vtab[i-1])
      /(vtab[i]-vtab[i-1]);
    return(ret);
}


float bessj0(float x)
{
    float ax,z;
    double xx,y,ans,ans1,ans2;

    if ((ax=fabs(x)) < 8.0) {
        y=x*x;
        ans1=57568490574.0+y*(-13362590354.0+y*(651619640.7
            +y*(-11214424.18+y*(77392.33017+y*(-184.9052456)))));
        ans2=57568490411.0+y*(1029532985.0+y*(9494680.718
            +y*(59272.64853+y*(267.8532712+y*1.0))));
        ans=ans1/ans2;
    } else {
        z=8.0/ax;
        y=z*z;
        xx=ax-0.785398164;
        ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4
            +y*(-0.2073370639e-5+y*0.2093887211e-6)));
        ans2 = -0.1562499995e-1+y*(0.1430488765e-3
            +y*(-0.6911147651e-5+y*(0.7621095161e-6
            -y*0.934935152e-7)));
        ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
    }
    return ans;
}
```

C1O0A.dyn

```
#include "\ALSIM\ALSIM.H"
#include "RTP.H"
#include <math.h>

#define DEBUG 0


/*

 This program performs the open loop control simulation
 Assumptions here are that initial states and disturbances are zero

 When this control law is applied both the states and derivatives
 of the states of the systems remain at zero if assumptions are true.


*/

double RO=2330.;              /* Wafer density  Kg/m^3              */
double Sigmaa=5.67e-8;        /* Stefan-Boltzmann, W/(m^2 K^4) */
double hh=6.35e-4;            /*      Wafer thickness, m            */
double m1=3.8317;
double m2=7.0156;
double R=76.2e-3;             /*       Wafer radius, m              */
double r1=0.;                 /*       1st sensor position          */
double r2=2.3e-2;             /*    2nd sensor position, m          */
double r3=4.6e-2;             /*    3rd sensor position, m          */

double G[4] [4]={                     /*       lamp ring rad. func.          */
    {111.,  111.  ,111.  ,111.  },
    {111.,  2808.  ,17043. ,20731. },
    {111.,  -1.22  ,2480. ,-615.  },
    {111.,  3908.  ,-3254. ,-2232. }
};
                                              /*          G^-1       */
double Gm1[4] [4]={
        {111.,  111.          ,111.       ,111.  },
        {111., 2.865508977703938e-5,  1.118539909154441e-4,  2.353308520406833e-4},
        {111., 9.148491266928918e-6,  3.318667114798136e-4, -6.469827556174777e-6},
        {111., 3.683463273570035e-5,  -2.879788900796406e-4,-2.655654630699686e-5},
 };


double Tbar[3]=                                 /***   T = Tbar[1] + Tbar2 * t   ***/
    {111.,300., 30.};

double t;
double reftemp,temp4,temp3;
```

```
/*
** User state derivative function.
*/

void derv(t, x, dxdt)
double t, *x, *dxdt;
{


 double S1=0,S2=0,S3=0,g1tilde,g2tilde,g3tilde;

 if(t<=10)
 {
    reftemp=Tbar[1]+Tbar[2]*t;
    temp4=reftemp*reftemp*reftemp*reftemp;
    temp3=reftemp*reftemp*reftemp;
  }

  if(t>10)
  {
    reftemp=600.;
    Tbar[1]=600.;
    Tbar[2]=0.;
  }



  /*******************    Open loop control functions    ******************/


    g1tilde=hh/2*Gm1[1][1]*(RO*c_heat(reftemp)*Tbar[2]+
         E(reftemp)*temp4);

    g2tilde=hh/2*Gm1[2][1]*(RO*c_heat(reftemp)*Tbar[2]+
         E(reftemp)*temp4);

    g3tilde=hh/2*Gm1[3][1]*(RO*c_heat(reftemp)*Tbar[2]+
         E(reftemp)*temp4);

    u[1]=g1tilde;
    u[2]=g2tilde;
    u[3]=g3tilde;


  /******************* **   RTP dynamic equations   ************************/

  dxdt[1]=-Tbar[2]+(1./(RO*c_heat(reftemp+x[1])))*(-E(reftemp+x[1])*
       (reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])
       +2/hh*(2808.*g1tilde+17043.*g2tilde+20731.*g3tilde+S1));


  dxdt[2]=(1./(RO*c_heat(reftemp+x[1])))*(
       (-(m1/R)*(m1/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
       (reftemp+x[1])*(reftemp+x[1]))*x[2]+
       +2/hh*(-1.22*g1tilde+2480.*g2tilde-615.*g3tilde+S2));
```

```
dxdt[3]=(1./(RO*c_heat(reftemp+x[1])))*(
      (-(m2/R)*(m2/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1]))*x[3]+
      +2/hh*(3908.*g1tilde-3254.*g2tilde-2232.*g3tilde+S3));



/*********************  Temperature Equations  *********************/

 y[1]= reftemp + x[1] + x[2]*bessj0(m1*r1/R) + x[3]*bessj0(m2*r1/R);

 y[2]= reftemp + x[1] + x[2]*bessj0(m1*r2/R) + x[3]*bessj0(m2*r2/R);

 y[3]= reftemp + x[1] + x[2]*bessj0(m1*r3/R) + x[3]*bessj0(m2*r3/R);



 if(DEBUG)  {
    printf("\tT   %lf\tu1  %lf\tu2 %lf\u3 %lf \n",g1tilde);

    while( !(kbhit()) );
    clrscr();
  }


}
```

C1O2A.dyn
written by: Maurizio Fulco

```
#include "\ALSIM\ALSIM.H"
#include "RTP.H"
#include <math.h>
#include "\ALSIM\TC\ARE.H"   /***      header file for ARE solver     ***/

#define DEBUG 0

#define alpha1        fpar[1]
#define alpha2        fpar[2]
#define alpha3        fpar[3]
#define T_riccati     fpar[5]
#define q             fpar[6]
#define r             fpar[7]



/*

   This program uses the riccati routine to implement a SDREF to estimate
   just the states of the system. Good convergence of the states is shown.
   Next program c1o1b will try to estimate both the states and disturbance
   parameters. In this program the disturbance parameters are estimated by the
   algorithm in the adaptive controller.

*/

double RO=2330.;                /*      Wafer density  Kg/m^3          */
double Sigmaa=5.67e-8;          /*  Stefan-Boltzmann, W/(m^2 K^4)      */
double hh=6.35e-4;              /*      Wafer thickness, m             */
double m1=3.8317;
double m2=7.0156;
double R=76.2e-3;               /*        Wafer radius, m              */
double r1=0.;                   /*      1st sensor position, m         */
double r2=2.3e-2;               /*      2nd sensor position, m         */
double r3=4.6e-2;               /*      3rd sensor position, m         */

 double G[4] [4]={              /*        lamp ring rad. func.         */
     {111.,  111.  ,111.  ,111.  },
     {111.,  2808.  ,17043. ,20731. },
     {111.,  -1.22  ,2480.  ,-615.  },
     {111.,  3908.  ,-3254. ,-2232. }
};

double Gm1[4] [4]={                             /*     G^-1     */
        {111.,  111.          ,111.       ,111.  },
        {111., 2.865508977703938e-5,  1.118539909154441e-4,  2.353308520406833e-4},
        {111., 9.148491266928918e-6,  3.318667114798136e-4, -6.469827556174777e-6},
        {111., 3.683463273570035e-5,  -2.879788900796406e-4,-2.655654630699686e-5},
 };
```

```
double Tbar[3]=                    /*   For reference temperature    */
     {111.,300., 30.};


double reftemp=0,temp4=0;
double P_S[4]={111., 0., 0., 0.};
double a[4];
double b[4];
double F_S[4];
double gtildemax[4]={111., 1., 1., 1.};
double S_est[4],v[4];


/*
** User state derivative function.
*/

void derv(t, x, dxdt)
double t, *x, *dxdt;
{

 int i,SOL_EXIST;
 double S1=100.,S2=-100.,S3=10.,g1tilde=0.,g2tilde=0.,g3tilde=0.;
 double lambda1=0.,lambda2=0.,lambda3=0.,lambda;
 double lambdamax=1.,lambdalow=0.,lamupp, lamlow;
 double temp,temvec;
 double alpha[4];

 int n,m;
 double **A,**B,*Q,*Rn,**S;
 static double Ka[4][4];                    /***    Kalman matrix    ***/

 static double t1;

 n=3;                                       /***    number of states   ***/
 m=3;                                       /***    number of outputs  ***/


 /**********************    Memory allocation    **************************/

 A=dmatrix(1,n,1,n);
 B=dmatrix(1,n,1,m);
 Q=dvector(1,n);                            /***    diagonal matrix   ***/
 Rn=dvector(1,m);                           /***    diagonal matrix   ***/
 S=dmatrix(1,n,1,n);                        /***     Riccati matrix   ***/

 /*********************    End Memory allocation   ***************************/


 /*******************  Begin cycle for ARE computation  *********************/

 if (t>=t1)  {
```

```
/*********************          A = A transpose          **********************/

   A[1][1]=(1./(x[7]*RO*c_heat(reftemp+x[7])))*(-E(reftemp+x[7])*
           (reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7]));

   A[1][2]=0;

   A[1][3]=0;

   A[2][1]=0;

   A[2][2]=(1./(RO*c_heat(reftemp+x[7])))*
           (-(m1/R)*(m1/R)*k(reftemp+x[7])-
           4.*E(reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7]));

   A[2][3]=0;

   A[3][1]=0;

   A[3][2]=0;

   A[3][3]=(1./(RO*c_heat(reftemp+x[7])))*(
       (-(m2/R)*(m2/R)*k(reftemp+x[7])-4.*E(reftemp+x[7])*(reftemp+x[7])*
           (reftemp+x[7])*(reftemp+x[7])));



   /*********************          B = C transpose          **********************/

   B[1][1] = 1;    B[1][2]=1;  B[1][3]=1;
   B[2][1] = bessj0(m1*r1/R) ;  B[2][2]=bessj0(m1*r2/R) ;
   B[2][3] = bessj0(m1*r3/R) ;  B[3][1] =bessj0(m2*r1/R) ;
   B[3][2] = bessj0(m2*r2/R) ;  B[3][3]=bessj0(m2*r3/R) ;



   /*************************          Q, R          ******************************/

   Q[1] = q;      Q[2] = q;      Q[3] = q;

   Rn[1] = r;      Rn[2] = r;      Rn[3] = r;


   /**********************    Computation of ARE     **********************/

   riccati(A,B,Q,Rn,n,m,S);

   /***************          K = S' * C' * R^(-1)          *********************/

   Ka[1][1]=(S[1][1]+S[2][1]*bessj0(m1*r1/R)+S[3][1]*bessj0(m2*r1/R))/r;
   Ka[1][2]=(S[1][1]+S[2][1]*bessj0(m1*r2/R)+S[3][1]*bessj0(m2*r2/R))/r;
```

```
    Ka[1][3]=(S[1][1]+S[2][1]*bessj0(m1*r3/R)+S[3][1]*bessj0(m2*r3/R))/r;
    Ka[2][1]=(S[1][2]+S[2][2]*bessj0(m1*r1/R)+S[3][2]*bessj0(m2*r1/R))/r;
    Ka[2][2]=(S[1][2]+S[2][2]*bessj0(m1*r2/R)+S[3][2]*bessj0(m2*r2/R))/r;
    Ka[2][3]=(S[1][2]+S[2][2]*bessj0(m1*r3/R)+S[3][2]*bessj0(m2*r3/R))/r;
    Ka[3][1]=(S[1][3]+S[2][3]*bessj0(m1*r1/R)+S[3][3]*bessj0(m2*r1/R))/r;
    Ka[3][2]=(S[1][3]+S[2][3]*bessj0(m1*r2/R)+S[3][3]*bessj0(m2*r2/R))/r;
    Ka[3][3]=(S[1][3]+S[2][3]*bessj0(m1*r3/R)+S[3][3]*bessj0(m2*r3/R))/r;

    t1=t1+T_riccati;

}

/**************        End cycle for ARE computation    ********************/


alpha[1] = alpha1;
alpha[2] = alpha2;
alpha[3] = alpha3;

if(t<=10)
{
    reftemp=Tbar[1]+Tbar[2]*t;
    temp4=reftemp*reftemp*reftemp*reftemp;
}

if(t>10)
{
    reftemp=600.;
    Tbar[1]=600.;
    Tbar[2]=0.;
}


if(t<16.)

  alpha[2] = (t-.1)*150;

if(t<=.1)

  alpha[2] = 0;


        /*********          Formulas for vector a          **************/

  a[1]=(0.5*hh*RO*c_heat(reftemp+x[1]))*(Gm1[1][1]*x[1]+
        Gm1[1][2]*x[2]+Gm1[1][3]*x[3]);

  a[2]=(0.5*hh*RO*c_heat(reftemp+x[1]))*(Gm1[2][1]*x[1]+
        Gm1[2][2]*x[2]+Gm1[2][3]*x[3]);

  a[3]=(0.5*hh*RO*c_heat(reftemp+x[1]))*(Gm1[3][1]*x[1]+
        Gm1[3][2]*x[2]+Gm1[3][3]*x[3]);
```

```
/*****************        Formulas for vector b        ******************/

F_S[1]=0.5*hh*RO*c_heat(reftemp+x[1])*Tbar[2]+0.5*hh*
      E(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])
      -S_est[1];

F_S[2]=0.5*hh*((m1/R)*(m1/R)*k(reftemp+x[1])+4.*E(reftemp+x[1])
      *(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1]))*x[2]-S_est[2];

F_S[3]=0.5*hh*((m2/R)*(m2/R)*k(reftemp+x[1])+4.*E(reftemp+x[1])
      *(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1]))*x[3]-S_est[3];


b[1]=Gm1[1][1]*F_S[1]+Gm1[1][2]*F_S[2]+Gm1[1][3]*F_S[3];

b[2]=Gm1[2][1]*F_S[1]+Gm1[2][2]*F_S[2]+Gm1[2][3]*F_S[3];

b[3]=Gm1[3][1]*F_S[1]+Gm1[3][2]*F_S[2]+Gm1[3][3]*F_S[3];


/*************        Calculation of lambda optimized to constraints     **********/

lamupp=lambdamax;
lamlow=lambdalow;

SOL_EXIST = 1;

for (i=1;i<=3;i++)
 {
      temp=b[i]/a[i];
      temvec=(b[i]-gtildemax[i])/a[i];
      if(a[i]>0.){
           if(lamlow<temvec)   lamlow=temvec;
           if(lamupp>temp)     lamupp=temp;
      }
      if(a[i]<0.){
           if(lamlow<temp)     lamlow=temp;
           if(lamupp>temvec)   lamupp=temvec;
      }
 }

 if(lamlow>lamupp) SOL_EXIST=0;

 lambda=lamupp;

 if(lambda<0) lambda=0;

 lambda1=lambda;
 lambda2=lambda;
 lambda3=lambda;
```

```
/***********          For feedback control equations          ************/

S_est[1] = (alpha[1]*x[1]+x[4])*SOL_EXIST;

S_est[2] = (alpha[2]*x[2]+x[5])*SOL_EXIST;

S_est[3] = (alpha[3]*x[3]+x[6])*SOL_EXIST;

P_S[1] = 0.5*hh*RO*c_heat(reftemp+x[1])*(Tbar[2]-lambda1*x[1])+
      0.5*hh*E(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])
       -S_est[1];

P_S[2] = 0.5*hh*RO*c_heat(reftemp+x[1])*(-lambda2*x[2])
      +0.5*hh*((m1/R)*(m1/R)*k(reftemp+x[1])
      +4.*E(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1]))*x[2]-S_est[2];

P_S[3] = 0.5*hh*RO*c_heat(reftemp+x[1])*(-lambda3*x[3])
      +0.5*hh*((m2/R)*(m2/R)*k(reftemp+x[1])
      +4.*E(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1]))*x[3]-S_est[3];


/*****************          Control functions          ****************/

   g1tilde=(Gm1[1][1]*P_S[1]+Gm1[1][2]*P_S[2]+Gm1[1][3]*P_S[3])*SOL_EXIST;

   g2tilde=(Gm1[2][1]*P_S[1]+Gm1[2][2]*P_S[2]+Gm1[2][3]*P_S[3])*SOL_EXIST;

   g3tilde=(Gm1[3][1]*P_S[1]+Gm1[3][2]*P_S[2]+Gm1[3][3]*P_S[3])*SOL_EXIST;


   u[1]=g1tilde;
   u[2]=g2tilde;
   u[3]=g3tilde;


/************          System and observers equations          ***************/

 dxdt[1]=-Tbar[2]+(1./(RO*c_heat(reftemp+x[1])))*(-E(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])
      +2./hh*(2808.*g1tilde+17043.*g2tilde+20731.*g3tilde+S1));


 dxdt[2]=(1./(RO*c_heat(reftemp+x[1])))*(
      (-(m1/R)*(m1/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1]))*x[2]+
      +2./hh*(-1.22*g1tilde+2480.*g2tilde-615.*g3tilde+S2));

 dxdt[3]=(1./(RO*c_heat(reftemp+x[1])))*(
      (-(m2/R)*(m2/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1]))*x[3]+
```

```
      +2./hh*(3908.*g1tilde-3254.*g2tilde-2232.*g3tilde+S3));

dxdt[4]=(-alpha[1]*(-Tbar[2]+(1./(RO*c_heat(reftemp+x[1])))*(-E(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])*(reftemp+x[1])
      +2./hh*(2808.*g1tilde+17043.*g2tilde+20731.*g3tilde+S_est[1]))))
      *SOL_EXIST;


dxdt[5]=(-alpha[2]*((1./(RO*c_heat(reftemp+x[1])))*(
      (-(m1/R)*(m1/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1]))*x[2]+
      +2./hh*(-1.22*g1tilde+2480.*g2tilde-615.*g3tilde+S_est[2]))))
      *SOL_EXIST;

dxdt[6]=(-alpha[3]*((1./(RO*c_heat(reftemp+x[1])))*(
      (-(m2/R)*(m2/R)*k(reftemp+x[1])-4.*E(reftemp+x[1])*(reftemp+x[1])*
      (reftemp+x[1])*(reftemp+x[1]))*x[3]+
      +2./hh*(3908.*g1tilde-3254.*g2tilde-2232.*g3tilde+S_est[3]))))
      *SOL_EXIST;

dxdt[7] =-Tbar[2]+(1./(RO*c_heat(reftemp+x[7])))*(-E(reftemp+x[7])*
      (reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7])*(reftemp+x[7])
      +2./hh*(2808.*g1tilde+17043.*g2tilde+20731.*g3tilde+S1))
      +Ka[1][1]*(x[1] + x[2]*bessj0(m1*r1/R) + x[3]*bessj0(m2*r1/R)-
      x[7] - x[8]*bessj0(m1*r1/R) - x[9]*bessj0(m2*r1/R))
      +Ka[1][2]*(x[1] + x[2]*bessj0(m1*r2/R) + x[3]*bessj0(m2*r2/R)-
      x[7] - x[8]*bessj0(m1*r2/R) - x[9]*bessj0(m2*r2/R))
      +Ka[1][3]*(x[1] + x[2]*bessj0(m1*r3/R) + x[3]*bessj0(m2*r3/R)-
      x[7] - x[8]*bessj0(m1*r3/R) - x[9]*bessj0(m2*r3/R));


dxdt[8]=(1./(RO*c_heat(reftemp+x[7])))*(
      (-(m1/R)*(m1/R)*k(reftemp+x[7])-4.*E(reftemp+x[7])*(reftemp+x[7])*
      (reftemp+x[7])*(reftemp+x[7]))*x[8]+
      +2./hh*(-1.22*g1tilde+2480.*g2tilde-615.*g3tilde+S2))
      +Ka[2][1]*(x[1] + x[2]*bessj0(m1*r1/R) + x[3]*bessj0(m2*r1/R)-
      x[7] - x[8]*bessj0(m1*r1/R) - x[9]*bessj0(m2*r1/R))
      +Ka[2][2]*(x[1] + x[2]*bessj0(m1*r2/R) + x[3]*bessj0(m2*r2/R)-
      x[7] - x[8]*bessj0(m1*r2/R) - x[9]*bessj0(m2*r2/R))
      +Ka[2][3]*(x[1] + x[2]*bessj0(m1*r3/R) + x[3]*bessj0(m2*r3/R)-
      x[7] - x[8]*bessj0(m1*r3/R) - x[9]*bessj0(m2*r3/R));


dxdt[9]=(1./(RO*c_heat(reftemp+x[7])))*(
      (-(m2/R)*(m2/R)*k(reftemp+x[7])-4.*E(reftemp+x[7])*(reftemp+x[7])*
      (reftemp+x[7])*(reftemp+x[7]))*x[9]+
      +2./hh*(3908.*g1tilde-3254.*g2tilde-2232.*g3tilde+S3))
      +Ka[3][1]*(x[1] + x[2]*bessj0(m1*r1/R) + x[3]*bessj0(m2*r1/R)-
      x[7] - x[8]*bessj0(m1*r1/R) - x[9]*bessj0(m2*r1/R))
      +Ka[3][2]*(x[1] + x[2]*bessj0(m1*r2/R) + x[3]*bessj0(m2*r2/R)-
      x[7] - x[8]*bessj0(m1*r2/R) - x[9]*bessj0(m2*r2/R))
      +Ka[3][3]*(x[1] + x[2]*bessj0(m1*r3/R) + x[3]*bessj0(m2*r3/R)-
      x[7] - x[8]*bessj0(m1*r3/R) - x[9]*bessj0(m2*r3/R));

 /********** End System and Observer equations  ******************/
```

```
/******************        Memory deallocation        **********************/

free_dmatrix(A,1,n,1,n);
free_dmatrix(B,1,n,1,m);
free_dvector(Q,1,n);
free_dvector(Rn,1,m);
free_dmatrix(S,1,n,1,n);


/*****************        Temperature Equations        ******************/

y[1]= reftemp + x[1] + x[2]*bessj0(m1*r1/R) + x[3]*bessj0(m2*r1/R);

y[2]= reftemp + x[1] + x[2]*bessj0(m1*r2/R) + x[3]*bessj0(m2*r2/R);

y[3]= reftemp + x[1] + x[2]*bessj0(m1*r3/R) + x[3]*bessj0(m2*r3/R);

y[4]=lambda;

y[5] = reftemp;

y[6] = Ka[3][3];

  if(DEBUG)  {
/*     printf("\tT   %lf\tu1  %lf\tu2 %lf\u3 %lf \n",g1tilde);*/

    while( !(kbhit()) );
    clrscr();
  }

}
```

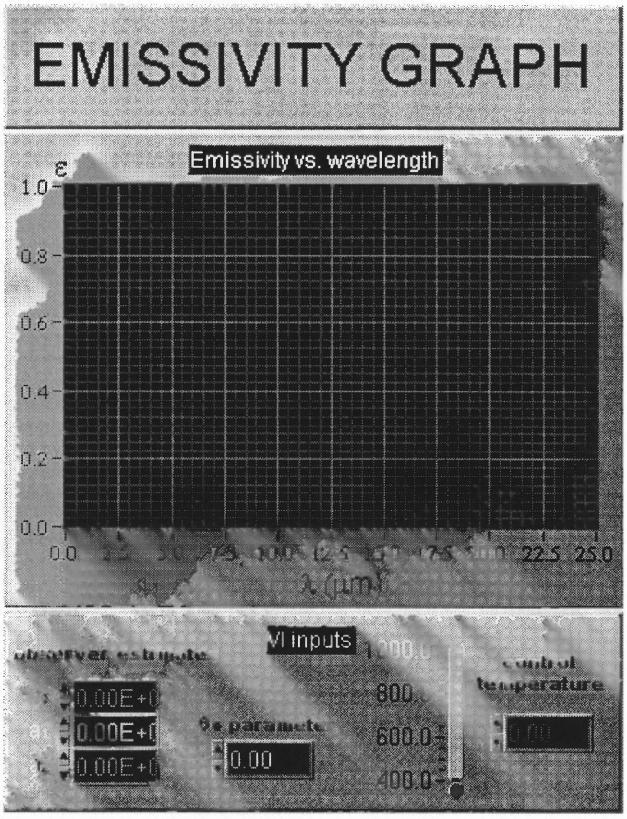
APPENDIX B

LabVIEW DOCUMENTATION

Emissivity.vi

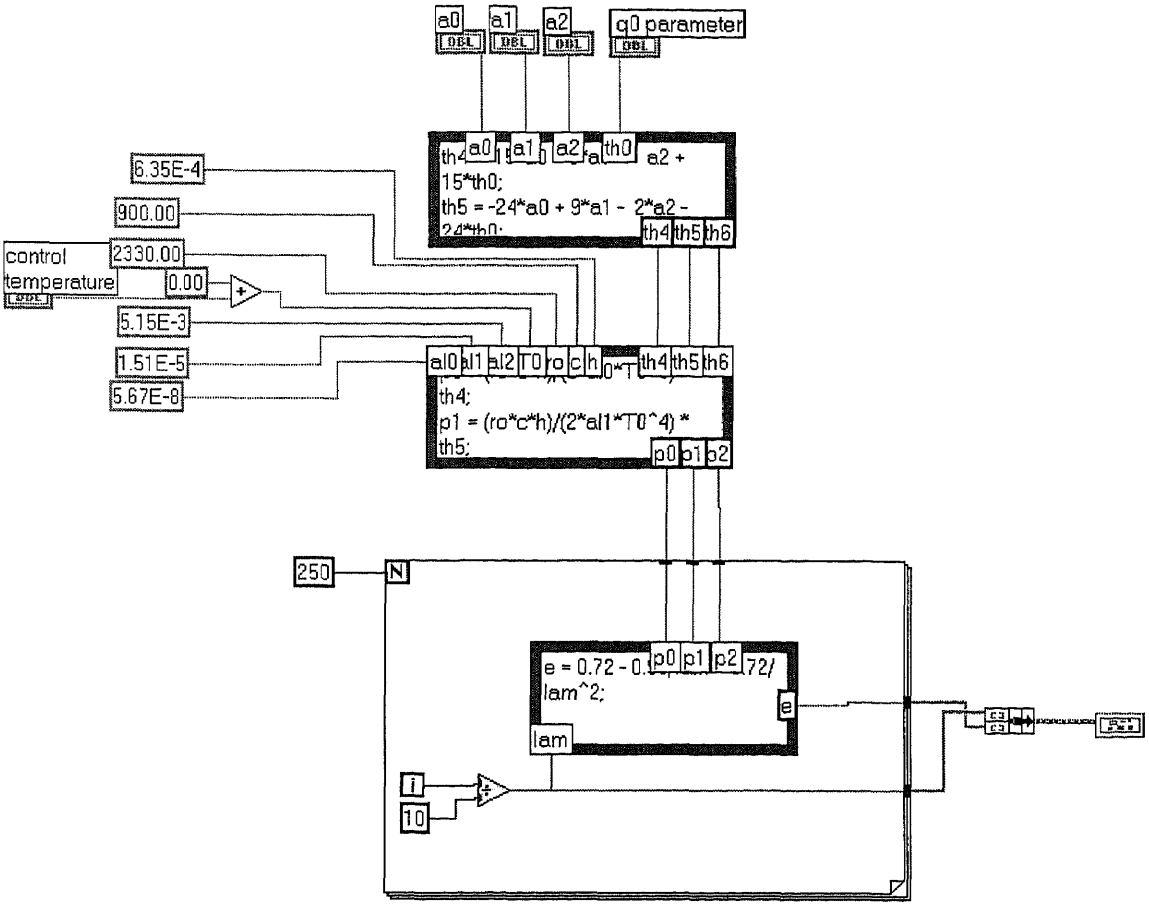
****

Front Panel



****

Block Diagram



****

Position in Hierarchy and Connector Pane

VI inputs:

- control temperature
- a0
- a1
- a2

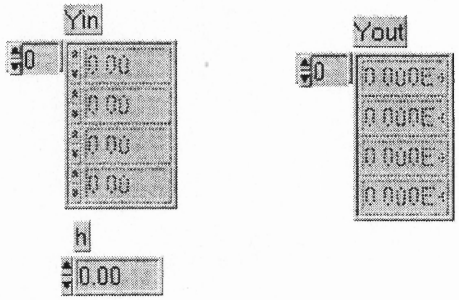
Controls and Indicators

- q0 parameter
- a0
- a1
- a2
- control temperature
- Emissivity

Runger Kutta.vi

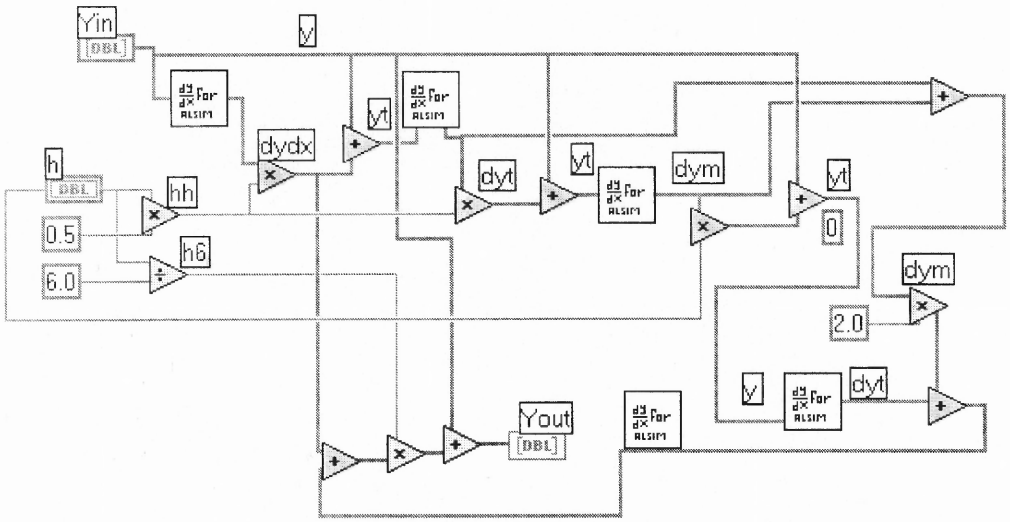
****

Front Panel



****

Block Diagram



****

VI inputs:
Yin
h
VI outputs:
Yout

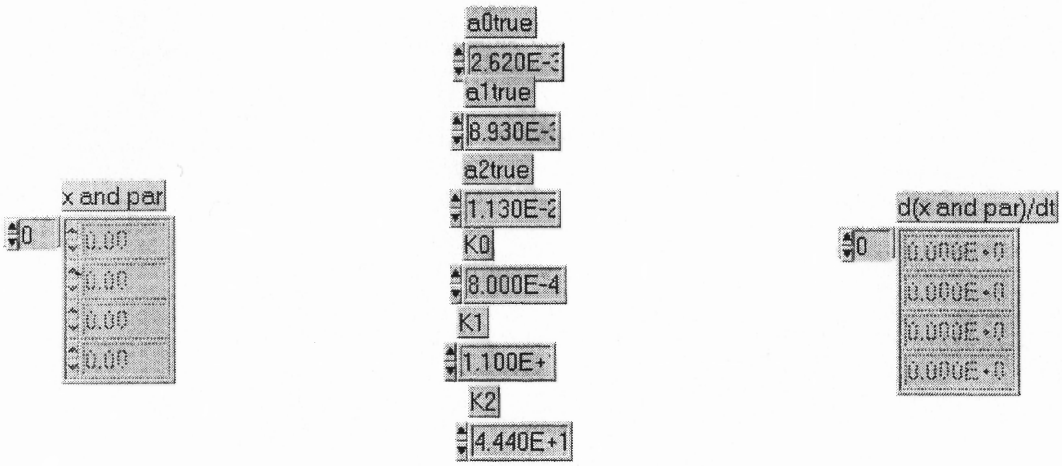
****

List of SubVIs;
dydx.vi

dydx.vi

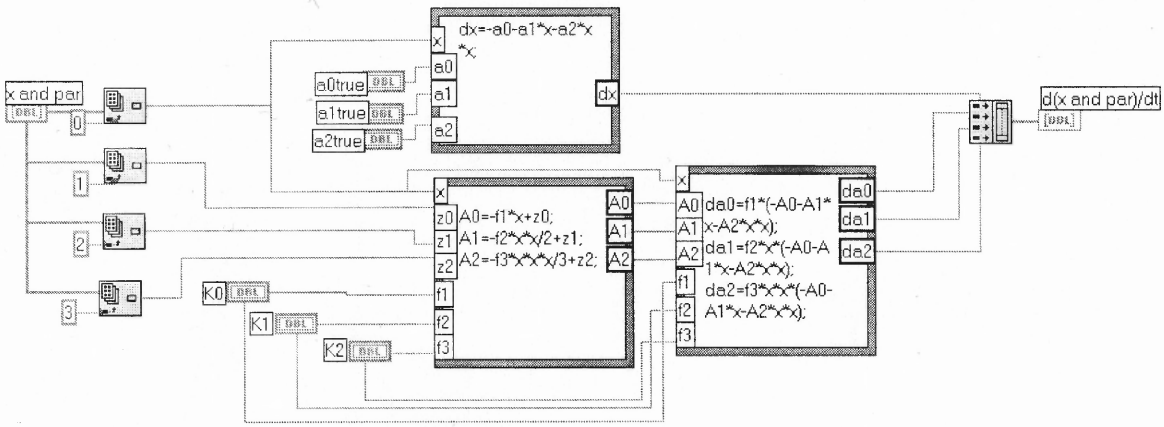
****

Front Panel



****

Block Diagram



****

Position in Hierarchy and Connector Pane

VI inputs:

x and par

VI outputs:

d(x and par)/dt

Controls and Indicators:

a0true

a1true

a2true

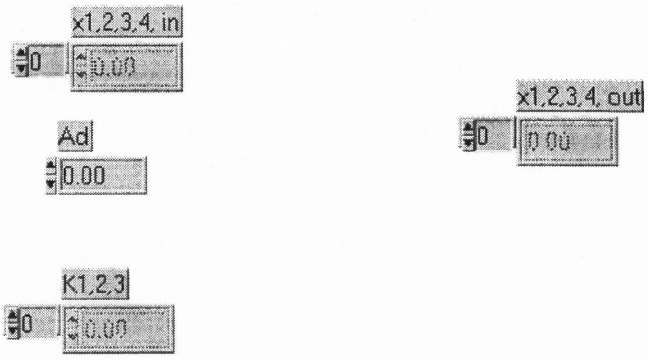
K0
K1
K2
x and par
d(x and par)/dt

****

Excitation.vi

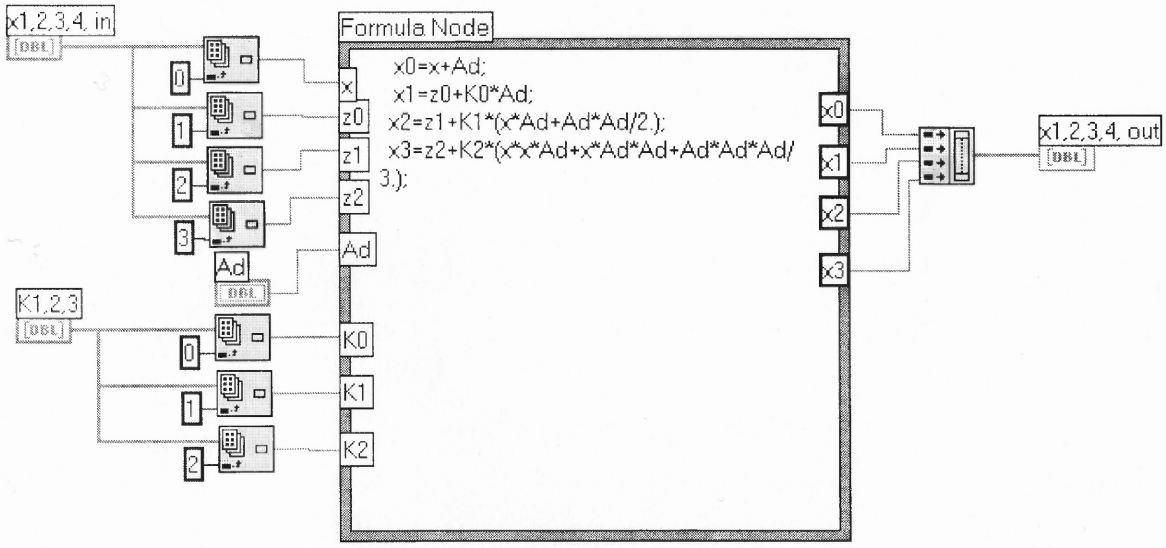
****

Front Panel



****

Block Diagram



****

Position in Hierarchy and Connector Pane

VI inputs:

x1,2,3,4,in
K1,2,3
Ad

VI outputs:

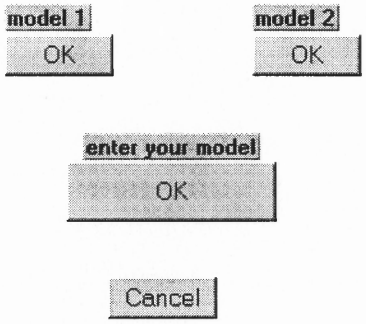
x1,2,3,4out

****

Select model.vi

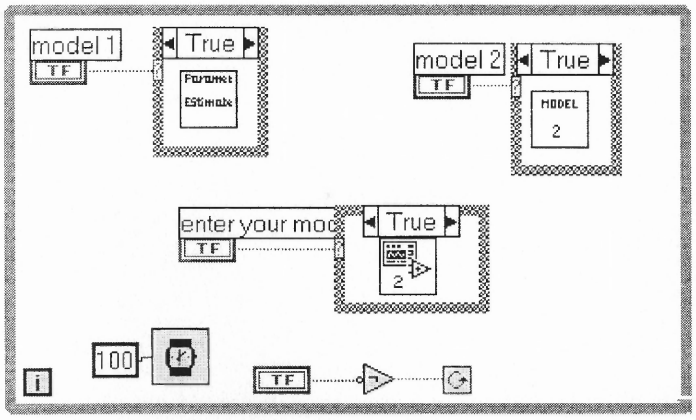
****

Front Panel



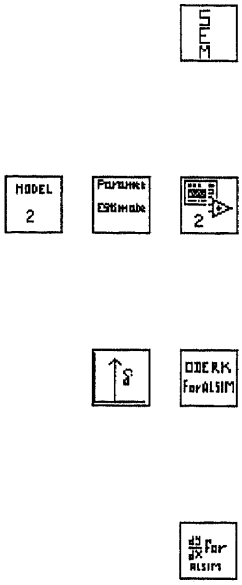
****

Block Diagram



****

Position in Hierarchy and Connector Pane



VI inputs:

Cancel

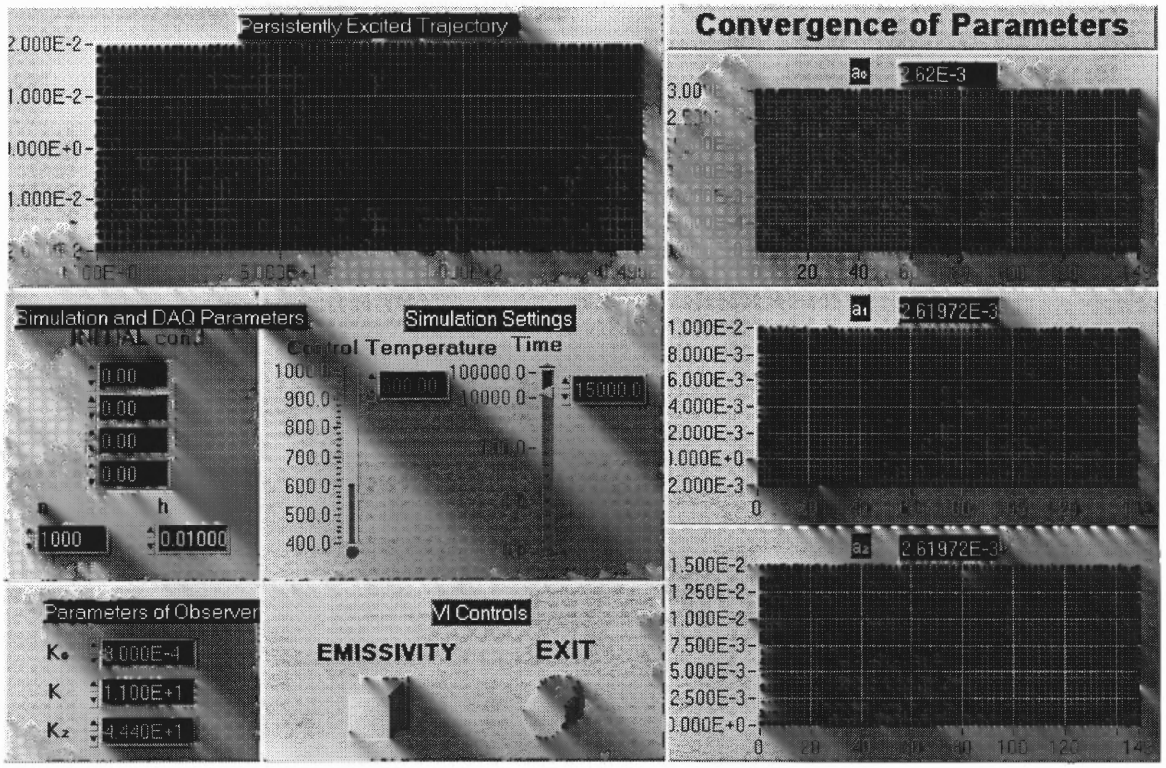
Control and Indicators

Cancel
Model1
Model2
Enter your model

Observer.vi

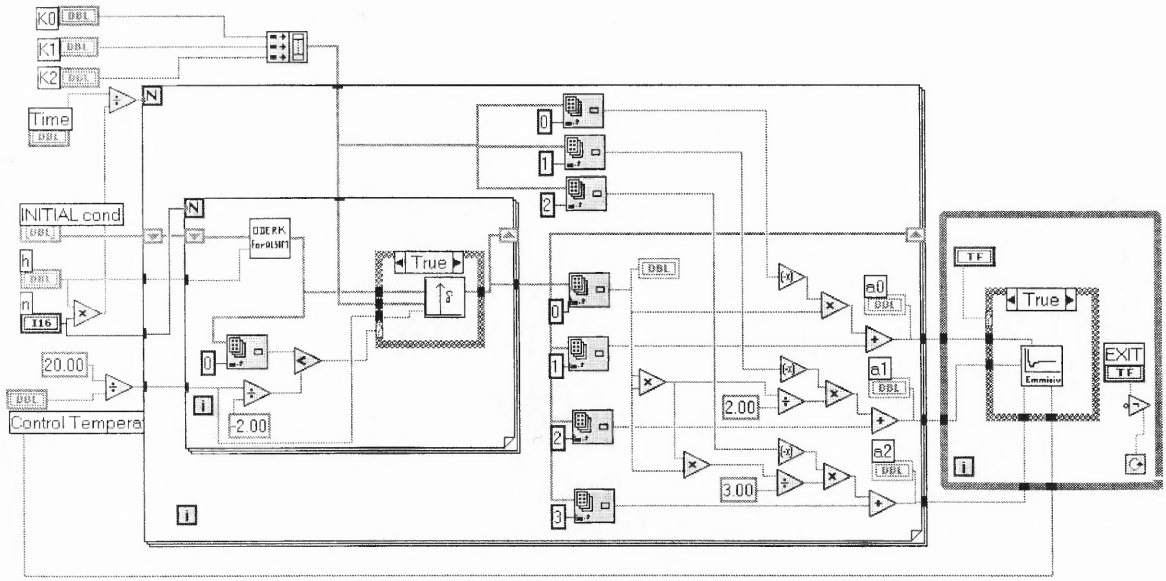
****

Front Panel



****

Block Diagram



Position in Hierarchy and connector pane

VI inputs:

h
n

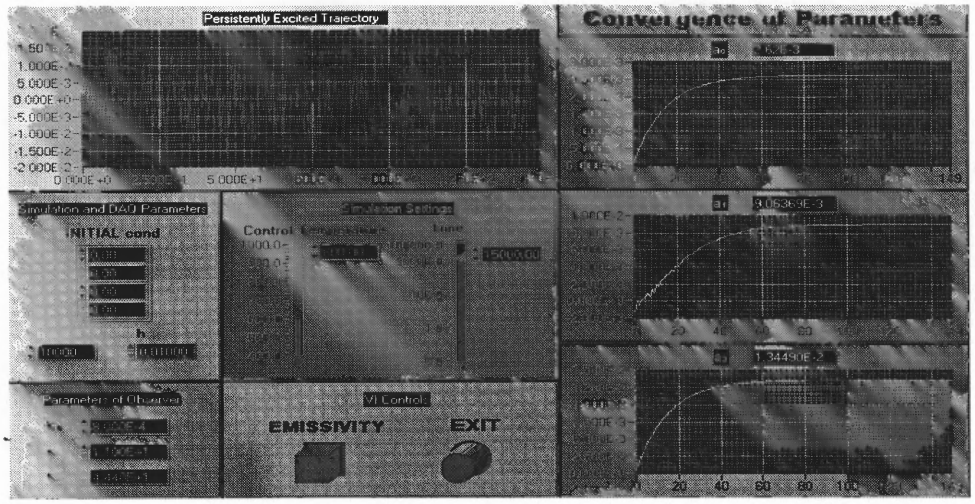
Controls and Indicators:

Initial Cond
h
Time
K0
K1
K2
Control Temperature
n
emissivity
EXIT
x
a0
a1
a2

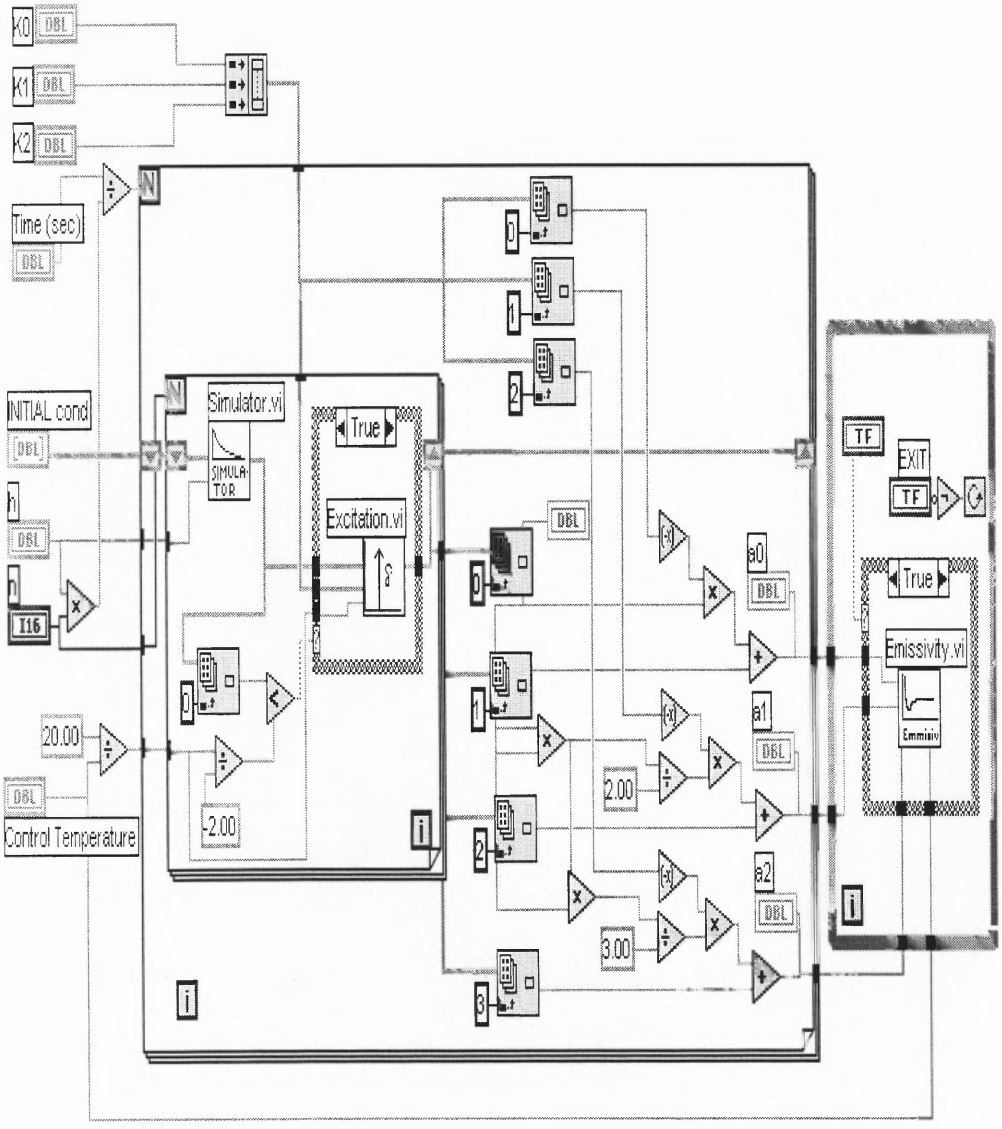
****

List of SubVIs

Runger Kutta.vi
Excitation.vi
Emissivity.vi



**** Block Diagram *****



REFERENCES

[1] Stephen A. Norman, "Wafer Temperature Control in Rapid Thermal Processing," Ph.D. Thesis, Dept. of Electrical Engineering, Stanford Univ., Stanford, CA, 1992.

[2] S. Belikov, B. Friedland, "Closed-loop adaptive control for Rapid Thermal Processing", in *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA, December 1995.

[3] F.R. Kessler, J. Schnell, Z. Naturforsch, Vol. 13a, p. 458, 1958.

[4] T. Sato, "Spectral Emissivity of Silicon", *Japanese Journal of Applied Physics*, Vol. 6, no.3, pp. 339-347, March 1967.

[5] P.J. Timans, "The Thermal Radiative Properties of Semiconductor", *Advances in Rapid Thermal and Integrated Processes* (ed. F. Roozeboom), Kluwer Academic Publishers, Dordrecht, pp 35-101, 1996.

[6] M. Fulco, O.L. Russo, S. Belikov, W. Kosonocky, "An Interactive System for Wafer Emissivity Estimation as Determined in an RTP Chamber",in *Proceedings of Rapid Thermal and Integrated Processes VI*, Material Research Society Conference, San Francisco, CA, April 1997.

[7] S. Belikov, D. Hur, B. Friedland, N.M. Ravindra, "Estimation of Emissivity of a Wafer in an RTP Chamber by a Dynamic Observer", in *Proceedings of Rapid Thermal and Integrated Processes V*, Material Research Society Conference, Pittsburgh, PA, 1996.

[8] D. Hur, "Real-Time Implementation of an Adaptive Control System for a 3-Zone Rapid Thermal Processing Station", MS Thesis, Dept. of Electrical Engineering, NJIT, Newark, NJ, October 1996.

[9] D. Luenberger,"An Introduction to Observers", *IEEE Transactions on Automatic Control*, AC-16, No. 6, pp. 596-602, 1971

[10] B. Friedland, "A Simple Non-Linear Observer for Estimating Parameters in Dynamic Systems", *Proceedings of The 12th IFAC Congress*, Sydney, Australia, July 18-23 1993.

[11] *LabVIEW User Manual,* National Instruments, 1996.

[12] *G Math Toolkit Reference Manual*, National Instruments, 1996.

[13] S. Belikov, H. Martynov, M. Kaplinsky, C. Manikopoulos, "Using Wavelength-Dependent Emissivity of Semiconductor Wafer to Model Heat Transfer in Rapid Thermal Processing Station", *Proceedings, IEEE Transactions on Semiconductor Manufacturing*, Vol. 8, No. 3, August 1995.

[14] B. Friedland, *Advanced Control Systems Design*, Prentice Hall, Englewood Cliffs, NJ, 1996.

[15] S. Belikov, H. Martynov, M. Kaplinsky, C. Manikopoulos, N. Ravindra, W. Kosonocky, "A Design Methodology for Configuration of Lamps in an RTP System", in *Proceedings of the 2nd International Rapid Thermal Processing Conference*, Monterey, CA, pp. 120-129, August 31-September 2 1994.

[16] L.C. Thomas , *Heat Transfer - Professional Version,* PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

[17] S. Belikov, M. Kaplinsky, and B. Friedland, "Parameter Estimation for Evaluating Ability of a Rapid Thermal Processing System to Maintain Uniform Temperature", in *Proceedings of the 4th IEEE Conference on Control Applications*, Albany, NY, pp.643-648, 1995.

[18] J.R. Cloutier, C.A. D'Souza, and C.P. Mracek, "Non linear regulation and non linear H_∞ control via the state dependent Riccati equation technique," in *Proceedings of the International Conference on Non Linear Problems in Aviation and Aerospace,* May 1996.

[19] C.P. Mracek, J.R. Cloutier, and C.A. D'Souza, "A new technique for non linear Estimation," in *Proceedings of the 5th IEEE Conference on Control Applications,* Sept. 1996

[20] D.P. DeWitt and G.D. Nutter, *Theory and Practice of Radiation Thermometry*, John Wiley & Sons Inc., New York, NY, 1988.

[21] P.J. Timans, "Temperature Measurement Strategies for Rapid Thermal Processing in Semiconductor Manufacturing," in *Proceedings of the 4th International Conference on Advanced Thermal Processing of Semiconductors,* Boise, ID, pp. 145-156, Sept. 1996.