119
92

MACROSCOPIC CONVECTION IN THE THIN-FILM PROCESSOR

by

Kim R. Hunter

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

MASTER OF SCIENCE

in

Chemical Engineering

APPROVED:

Peter R. Rony
P. R. Rony

Kenneth Konrad
K. Konrad

H. A. McGee

November, 1988

Blacksburg, Virginia

MACROSCOPIC CONVECTION IN THE THIN-FILM PROCESSOR

by

Kim R. Hunter

Committee Chairman: Dr. Peter R. Rony

Chemical Engineering

(ABSTRACT)

The thesis explores the proposal that macroscopic fluid convection in thin-film processors may be adequately represented by simple linear deterministic models. In addition, it examines the suggestion that the models themselves provide a useful tool in the search for a generalizable 'intrinsic' process heat transfer film coefficient, ie., one that includes the effects of axial dispersion of the process fluid. Such a parameter would be helpful in the design and scale up of thin-film equipment.

The following approach was used to investigate this proposal: first, experimental fluid residence time distributions were obtained, over a range of operating conditions, using an industrial pilot plant thin-film processor. The experimental data were used to select an appropriate linear fluid flow model for the process. The model parameters were evaluated over this range using

frequency response techniques. These models were subsequently incorporated into a numerical heat transfer simulation of the thin-film processor. Careful matching of the pilot plant transient temperature responses to those predicted by the simulation yielded the sought after intrinsic (dispersion corrected) heat transfer film coefficients for the processor.

Thin-film processor size:	0.12 m^2
Process feed rate:	27-130 kg hr^{-1}
Rotor speed:	5-8.8 m s^{-1}
Intrinsic process heat transfer film coefficient:	190-770 W m^{-2} K^{-1}

Acknowledgements

First, and foremost, I must acknowledge the contribution of the Luwa Corporation (Charlotte, N.C.) for supporting this work with facilities and personnel. In particular, I would like to extend my appreciation to Bill Glover and Howard Freese for initially 'signing on' to the project, and providing steadfast executive support throughout the work; and to Ardell Stewart and George Morell for continuous assistance in scheduling, equipment modification, and the innumerable details of pilot plant operations. I experienced, first hand, the professionalism and capability for which Luwa Corporation has become renowned in the chemical process industry.

I must also take this opportunity to extend heart felt appreciation to the members of my thesis committee: Dr. Rony for serving as my chairman, Dr. Konrad (now of Exxon Research and Engineering) who was my advisor throughout the work described in this thesis, and Dr. McGee for his guidance and encouragement during my coursework. Also, prior to my thesis work, I received immeasurable assistance from Dr. Y. A. Liu. His encouragement and active mentoring over a period of two full years rescued my program on more than one occasion. Dr. Liu never allowed the option that I might fail or abandon my program in discouragement; and as a result, neither did I.

Finally, I want to give special thanks to Ms. Mee Ying Wong whose empathy, toughness, organizational talent, and love, drove this work forward against the forces of entropy.

K R H Durham, N. C. 3/31/1988

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Area (m^2)
A	Matrix scaling coefficient (section 3.2)
B	Matrix scaling coefficient (section 3.2)
C	Matrix scaling coefficient (section 3.2)
C	Concentration of tracer (kg m^{-3})
C_{in}	Concentration of tracer entering the system (kg m^{-3})
C_{out}	Concentration of tracer leaving the system (kg m^{-3})
C_i	Concentration of tracer in the i'th lumped parameter model stage (kg m^{-3})
C_N	Concentration of tracer in the final (N'th) model stage (kg m^{-3})
$\bar{c}$	Laplace transformed tracer concentration
$\bar{c}_{in}$	Laplace transformed concentration of tracer entering the system
$\bar{c}_{out}$	Laplace transformed concentration of tracer leaving the system
$\bar{c}_N$	Laplace transformed concentration of tracer in the N'th model stage
C_p	Specific heat capacity of the process fluid (J kg^{-1} K^{-1})
C_w	Specific heat capacity of the heat transfer wall (J kg^{-1} K^{-1})
D	Matrix scaling coefficient (section 3.2)
D	Dead time (s) (section 3.4)
D	Axial dispersion coefficient (m^2 s^{-1}) (section 3.1)
Dt	Dead time (s) (Table 9.1)

E	Matrix scaling coefficient (section 3.2)
F	Matrix scaling coefficient (section 3.2)
f	Frequency (Hz) (chapter 3)
f	Subscript denoting the process fluid, e.g. M_f, Q_f, T_f
G(s)	The Laplace transform of the transfer function for the model without dead time
G'(s)	The Laplace transform of the transfer function for the model with dead time
h	Heat transfer film coefficient (W m^{-2} K^{-1})
i	A subscript denoting an individual stage in a lumped parameter model, e.g. C_i, P_i, Q_i, T_i
I	A subscript denoting the process fluid side (inside) of the heat transfer wall, e.g. h_I
j	square root of -1
k	Thermal conductivity (W m^{-1} K^{-1})
k	The frequency component index in the discrete Fourier transform (section 3.3)
K	Temperature in degrees Kelvin
K	Matrix scaling coefficients (section 3.2)
L	Characteristic axial dimension (section 3.1)
L	The number of sampled data points in the discrete Fourier transform (section 3.3)

M	Mass (kg)
M_f	Process fluid mass in a generalized lumped parameter model stage (kg)
M_w	Heat transfer wall mass in a generalized lumped parameter model stage (kg)
n	Summation index in the discrete Fourier transform
N	Number of equilibrium stages in a lumped parameter model
N	Subscript denoting the final stage in the system, e.g. C_N, $\bar{c}_N$, T_N
O	Subscript denoting the steam heating side (outside) of the heat transfer wall, e.g. h_O, T_O
P	Power input from rotor (W)
P_i	Power input from rotor in the i'th model stage (W)
q	Volumetric flow rate in a generalized lumped parameter model stage ($m^3\ s^{-1}$)
q_{in}	Volumetric flow rate into the system ($m^3\ s^{-1}$)
q_{out}	Volumetric flow rate out of the system ($m^3\ s^{-1}$)
Q	Heat flux across the heat transfer wall ($W\ m^{-2}$)
Q_{fi}	Heat flux form the heat transfer wall into the process fluid in the i'th model stage ($W\ m^{-2}$)
Q_{wi}	Heat flux from the heating steam into the heat transfer wall in the i'th model stage ($W\ m^{-2}$)
Q_x	Heat flux at position x inside the heat transfer wall ($W\ m^{-2}$)

Q_{av} Average heat flux ($W\ m^{-2}$)

Q_{inst} Instantaneous heat flux ($W\ m^{-2}$)

Q Net heat transfer rate (W) (section 2.3)

$s = j2\pi f$ The Laplace transform independent variable [frequency (s^{-1})]

t Time (s)

t_c Dwell time used in penetration analysis (s)

T Process fluid temperature in a generalized lumped parameter model stage (^{o}C)

T_{bulk} Temperature in the bulk process fluid (^{o}C) for the penetration model (section 1.7)

T_i Process fluid temperature in the i'th model stage (^{o}C)

T_{in} Process fluid inlet temperature (^{o}C)

T_I Inside (process side) heat transfer wall temperature (^{o}C)

T_{last} The last fluid temperature calculated for a model stage by the numerical heat transfer simulation

T_N Process fluid temperature in the N'th lumped parameter mode stage (^{o}C)

T_O	Outside (heating side) heat transfer wall temperature (^oC)
T_{out}	Process fluid outlet temperature (^oC)
T_w	Heat transfer wall internal temperature in a generalized lumped parameter model stage (^oC)
T_{wall}	Temperature at the fluid/wall interface (^oC) for the penetration model (section 1.7)
T_{wi}	Heat transfer wall internal temperature in the i'th model stage (^oC)
T_{wN}	Internal heat transfer wall temperature in the N'th model stage (^oC)
$u = L/\bar{\tau}$	Characteristic mean axial velocity in the distributed parameter flow model ($m\ s^{-1}$)
u	Variable of integration in equation [1.2]
U	Overall heat transfer coefficient ($W\ m^{-2}\ K^{-1}$)
V	Volume of a single stage in a lumped parameter model (m^3)
V	Conductivity transducer output (volts) (section 8.5)
w	Mass flow rate ($kg\ hr^{-1}$); or a subscript referring to the heat transfer wall, e.g. T_w
x	Dimension perpendicular to the heat transfer wall (m)
x	Subscript denoting internal position on the heat transfer wall e.g. Q_x

$Y_{in}(t)$ System input variable, used in the system state equation

$Y_{out}(t)$ System output variable, used in the system state equation

y_{calc} System output variable in the frequency domain (calculated from the lumped parameter model)

y_{obs} System output variable in the frequency domain (measured experimentally)

z Axial coordinate in the distributed parameter flow model

$Z = z/L$ Dimensionless axial coordinate

α Thermal diffusivity ($m^2\ s^{-1}$)

β Quadratic coefficients for the SSE response surface equation (section 6.4)

β_0 constant coefficient

β_1 coefficient scaling h_O

β_2 coefficient scaling h_I

β_{11} coefficient scaling $h_O h_O$

β_{12} coefficient scaling $h_O h_I$

β_{22} coefficient scaling $h_I h_I$

Δf	Sampling interval in the frequency domain (Hz)
Δt	Sampling interval in the time domain (s) (section 3.3)
Δt	Time increment for the numerical heat transfer simulation program (s) (section 6.3)
ΔT	Generalized temperature difference between the process and heating fluids (K)
ΔT_1	Terminal temperature difference at the high temperature end in the LMTD model (section 2.4)
ΔT_2	Terminal temperature difference at the low temperature end in the LMTD model (section 2.4)
ΔT_{LM}	Log Mean Temperature Difference between the high and low temperature ends in the LMTD model (section 2.4)
Δx	Incremental distance in the direction of heat flow within the heat transfer wall (m)
ρ	Density of the process fluid; or of the metal heat transfer wall (kg m^{-3})
τ	Mean residence time in a single stage of a lumped parameter flow system (s)
$\bar{\tau}$	Mean residence time in the distributed parameter flow model (s)
θ_{exp}	Experimental mean residence time (s) (Table 9.1)

CHAPTER ONE

Introduction

1.1 The Thesis

The work which follows is based on a proposal that macroscopic fluid convection in thin-film processors can be adequately represented by simple, linear, deterministic models. Further, it is suggested that the models themselves provide a useful tool in the search for a generalizable 'intrinsic' process heat transfer film coefficient. Such a parameter would be very helpful in attempts to automate the design and scale up of thin-film equipment.

1.2 The Strategy

The following approach was used to investigate this proposal:
First, experimental fluid residence time distributions (more correctly termed residence time probability density functions, Lapidus and Seinfeld (1974, Chapter 7) were obtained over a range of operating conditions using an industrial thin-film processor pilot plant. The experimental data were used to determine an appropriate linear fluid flow model for the process. The model parameters were evaluated over these operating conditions using frequency response techniques. These models were subsequently incorporated into a numerical heat transfer simulation of the thin-film processor. This is

shown schematically in Figure 1.1. A careful matching of the pilot plant transient temperature responses to those predicted by the simulation yielded the intrinsic (dispersion corrected) heat transfer film coefficients for the process.

1.3 Research Objective

The principal objective was to demonstrate a method to obtain intrinsic film coefficients for thin-film equipment. The key elements of the method are:

- Representation of bulk fluid convection by a linear lumped-parameter model.

- Incorporation of the lumped-fluid flow model in a numerical simulation of heat transfer in a thin-film processor.

- Determination of the parameter for thermal convection at the fluid/wall interface (the film coefficient) by adjusting the simulation transient temperature response to match the experimental response.

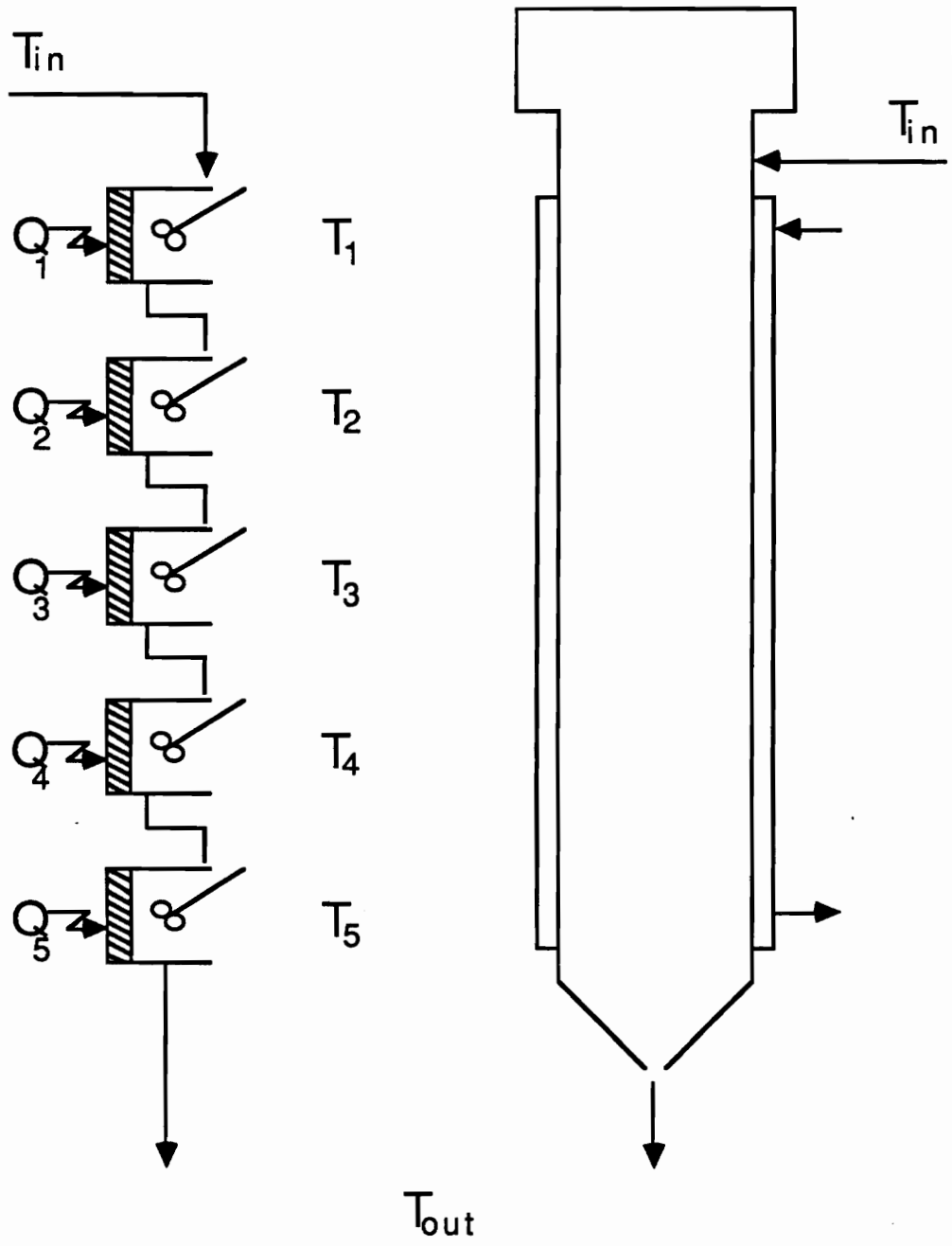


Figure 1.1. A schematic model for the numerical heat transfer simulation.

1.4 Presentation Of The Research

The work is presented in three parts. The first part, Chapters 3-5, formulates the linear state models for fluid convection in the thin-film processor. The requisite analytic and experimental techniques for model verification and parameter estimation are covered here. The second part, Chapters 6-8, integrates the model state equations into a comprehensive numerical simulation of heat transfer in the thin-film processor. Details of the heat transfer experiments are given here. Data acquisition and processing methods are also treated. In the third part, Chapters 9-10, the research results are presented and interpreted. The results are, principally, the intrinsic heat transfer film coefficients obtained from experimental transient temperature data. The intrinsic heat transfer film coefficient is defined in this work as: *the value of the process heat transfer film coefficient after the inclusion of axial mixing and dispersion effects.*

1.5 The Intended Application

Assume the effects of dispersion could be uncoupled from conventionally measured heat transfer film coefficients and the dispersion incorporated, instead, directly into a state model of the process. The film coefficients thus obtained would remain strong functions of *imposed* process conditions such as feed rate and rotor speed, but could be largely *independent* of *passive* factors accounting for gross axial fluid convection such as unit *size* and *geometry*.

This would allow a characteristic film heat transfer coefficient evaluated for one unit at specific operating conditions to be combined with the fluid convection model of another (perhaps larger) unit yielding a simulation capable of predicting the heat transfer performance of the new unit at those operating conditions. The potential of such methods for automating the design and scale up of thin-film equipment is evident.

1.6 A Survey of Reported Research

In the literature, the terms 'thin-film', and 'scraped-surface' are often used interchangeably. However, in this work, the term 'thin-film processor' is restricted to mean those units that incorporate a fixed clearance between the rotor blades and the heat transfer surface. A 'scraped-surface' unit is equipped with hinged blades which contact the heat transfer surface. The two units have many similarities, but there is no guarantee that the macroscopic convection is the same in both cases.

The published work on heat transfer in thin-film processors can be roughly grouped into two areas: 1. correlation of experimental results by dimensional analysis and, 2. microscopic modeling of heat transfer using the 'penetration theory'. No macroscopic or 'state' models of thin-film heat transfer were found in the literature. However, Asbjornsen (1961) analyzed fluid convection in falling films, and Calu and Lameloise (1988) modeled fluid convection in climbing-film evaporators, using state models and frequency response techniques.

A concise survey of the work on dimensional analysis is given by Maingonnat and Corrieu (1986a). They point out the traditional weakness in theoretical models: that the required simplifications do not lead to good agreement with experimental values. Conversely, empirical models are restricted to a particular unit geometry and set of operating conditions.

Most theoretical analyses (Kool, 1958; Harriot, 1959; Bott and Azoory, 1969; Woschitz, 1983) begin with a differential momentum, and energy, balance on an element of fluid at the heat transfer surface. The resulting differential equations are then integrated over the entire unit. All of the theoretical investigators used 'penetration theory' in their model development. Penetration theory is a mathematical representation of fluid interaction with the heated surface. An element of fluid contacts the surface and remains there for a short time. During the contact period, heat is transferred into the element by conduction. If contact times are short, the heat transferred during the contact period can be calculated using a 'conduction into a semi-infinite medium' model. At the end of the period, the fluid element is swept from the surface back into the bulk. In penetration models, conduction into the fluid elements alone accounts for transfer of heat from the wall into the bulk. The main obstacle in applying penetration theory is the uncertainty concerning the time the fluid element 'dwells' on the surface. Penetration theory has been especially attractive in modeling thin-film equipment because, ideally, the dwell time is equivalent to (or directly related to) the frequency^{-1} of the rotor blades. Kool

(1958) proposed such a model for scraped-surface heat exchangers, a model that served as a basis for numerous subsequent investigations. Penney and Bell (1967) provide an excellent review of this early work.

Harriot (1959) derived an expression for the inside film heat transfer coefficient based on Kool's model. However, the heat transfer coefficients predicted by this expression were consistently higher than those observed experimentally. One line of investigation (Trommelen et al., 1971) maintained that the discrepancy resulted from incomplete radial mixing of fluid elements in the bulk. Elements were being returned to the wall before they could give up their energy to the bulk, thus lowering both the mean temperature difference at the wall and the apparent heat transfer coefficient. Other workers (Bott, Azoory, and Porter, 1968; Maingonnet and Corrieu, 1986b) also believed that the discrepancy resulted from a reduction in the temperature difference at the wall. However, they maintained that this was caused by 'axial dispersion' rather than incomplete radial mixing.

The absence of an accurate model to estimate fluid temperature at the heat transfer surface prompted attempts to measure the 'local' energy flux at discrete points on the heat transfer wall. Bott and Azoory (1969) constructed a laboratory thin-film device with a thermally isolated heat transfer path in the wall. This 'flux meter' allowed determination of local fluxes. More recently, Miyashita and Hoffman (1978a, 1978b) and Woschitz (1983) have obtained instantaneous, point heat transfer coefficients using electrochemical methods. The charge flux of a conductive process fluid was

measured electrochemically at the wall by a point electrode. The mass transfer coefficient, calculated from the charge flux, can be converted to a heat transfer coefficient using an appropriate heat/mass transfer analogy for turbulent flow. However, it can be shown (Chap. 10) that film coefficients obtained by these methods do not capture the effects of axial dispersion.

1.7 Derivation of the Penetration Model

Over the last 25 years, the focus of heat transfer research in thin-film/scraped-surface process equipment has been the rationalization of process heat transfer coefficients that were determined experimentally (from log-mean temperature difference calculations, as discussed in section 2.5) with the values predicted by penetration theory. Although the penetration model is not used in the work presented here, a review of the formulation of 'penetration' heat transfer film coefficients is appropriate for historical reference.

Starting with the one-dimensional heat conduction equation (Wochitz, 1983):

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad ; \quad \alpha = \frac{k}{\rho C_p} \qquad [1.1]$$

and boundary and initial conditions:

$t < 0 \quad ; \quad 0 < x < \infty \; ; \; T = T_{bulk}$

$t > 0 \quad ; \quad x = 0 \quad ; \quad T = T_{wall}$

$t > 0 \quad ; \quad x \to \infty \quad ; \quad T = T_{bulk}$

where:

k = fluid thermal conductivity
ρ = fluid density
C_p= fluid heat capacity
x = dimension perpendicular to the heat transfer wall
T_{bulk}= temperature of the bulk fluid
T_{wall}= temperature at the heat transfer wall

The above system can be solved by Fourier transform methods (Duchateau and Zachmann, 1986) yielding:

$$T = T_{wall} - (T_{wall} - T_{bulk}) \frac{2}{(\pi)^{1/2}} \int_0^{\frac{x}{2(\alpha t)^{0.5}}} e^{u^2} du \qquad [1.2]$$

where 'u' is a variable of integration

Applying the definition of the error function to equation [1.2]

$$T = T_{wall} - (T_{wall} - T_{bulk}) \, \mathrm{erfc}\left[\frac{x}{2 (\alpha t)^{1/2}}\right] \qquad [1.3]$$

The instantaneous energy flux can be obtained by differentiating [1.3] with respect to x at the wall:

$$Q_{inst} = -k \left.\frac{dT}{dx}\right|_{x=0}$$

$$= k\,(T_{wall} - T_{bulk}) \left[\frac{1}{\pi\,\alpha\,t}\right]^{1/2} \qquad [1.4]$$

An average value for the flux, Q_{av}, is obtained from the integral mean value of equation [1.4] over the characteristic penetration time t_c. In a thin-film processor, t_c is the dwell time at the heat transfer wall or the time between successive sweeps of the rotor blades.

$$Q_{av} = \frac{1}{t_c} \int_o^{t_c} Q_{inst}\, dt$$

$$= 2k\,(T_{wall} - T_{bulk}) \left[\frac{1}{\pi\,\alpha\,t_c}\right] \qquad [1.5]$$

Recalling the definition of the heat transfer film coefficient 'h':

$$Q_{av} = h\,(T_{wall} - T_{bulk}) \qquad [1.6]$$

Substitute [1.5] into [1.6] to obtain the heat transfer film coefficient from penetration theory,

$$h = \frac{2}{(\pi)^{1/2}} \left[\frac{k\,\rho\,C_p}{t_c}\right]^{1/2} \qquad [1.7]$$

Note that the penetration theory does not consider the influence of fluid momentum transport, viscosity, or the velocity field in its prediction of heat transfer coefficients in thin-film equipment.

CHAPTER TWO

Background

2.1 A General Description Of The Thin-Film Processor

Thin-film processing is a unit operation employed for heat and/or mass transfer. One distinguishing feature of the thin-film processor is its low liquid inventory and corresponding low fluid residence time. These attributes are achieved through the mechanical action of a precisely machined longitudinal rotor. The configuration of a typical thin-film processor is shown schematically in Figure 2.1. The rotor and shell are machined to a tolerance of ± 1/2 mm. A fixed clearance between the rotor and shell of 1 to 5 mm is typically specified for industrial units. The rotor is mechanically stabilized by bearings at the upper and lower ends. In operation, the rotor is driven at a linear tip speed of 5 to 10 m/s. The process fluid is introduced at the top of the shell and distributed uniformly about its circumference in a thin film. This film is vigorously agitated by the rotor as it travels down the inner wall of the shell. A noteworthy feature of the flow regime is the propagation of a leading 'bow wave' at the front of each traveling rotor blade, Figures 2.1 and 2.2. The bow wave, or 'fillet' as it is often referenced in the literature (Kern and Karakas, 1959), is of primary importance in determining the properties of thin-film processors.

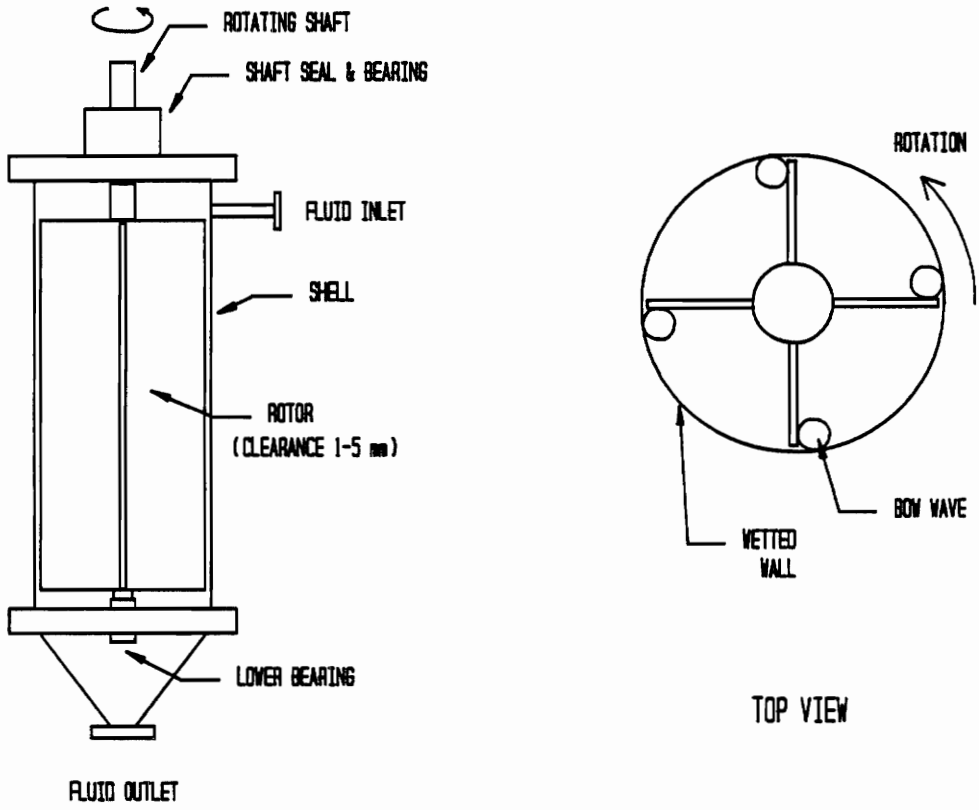


Figure 2.1. A highly schematic layout drawing of a thin-film processor.

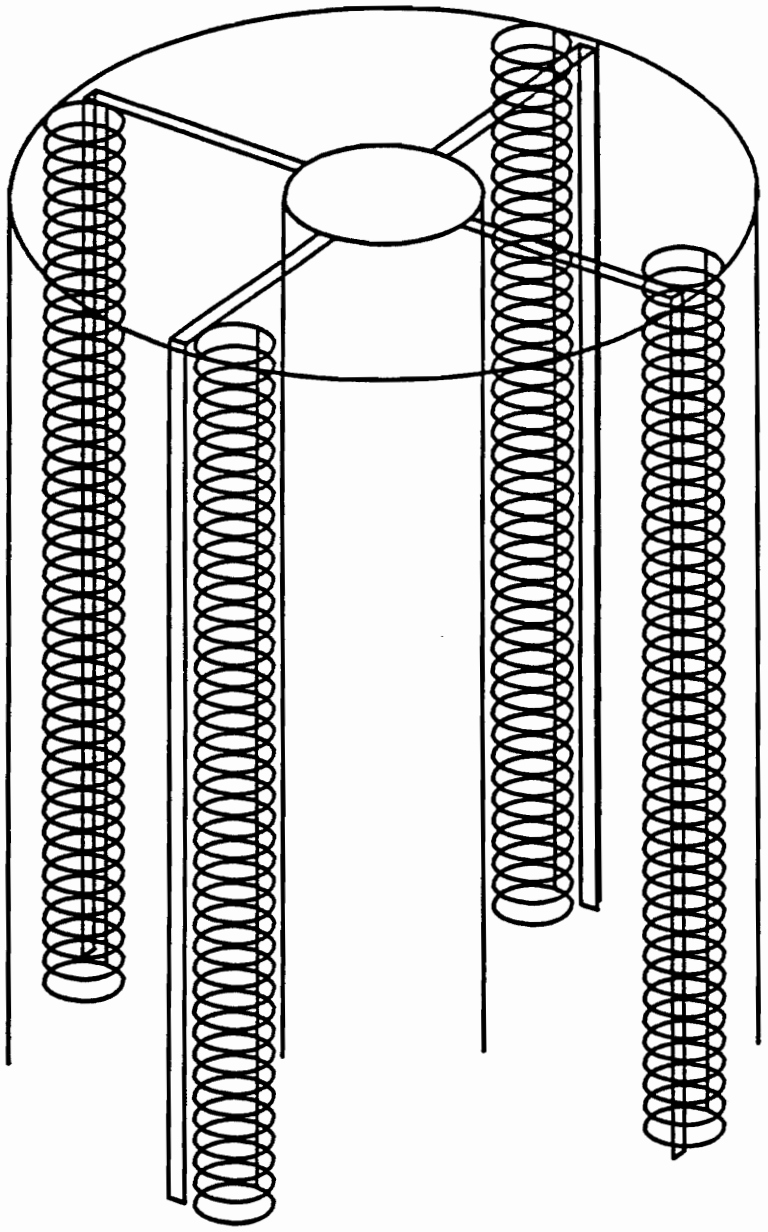


Figure 2.2. A visualization of the fluid bow wave. 'Fillets' of fluid are swept ahead of the advancing rotor blades.

These properties include high mass and heat transfer coefficients, low residence time, and a narrow residence time distribution when compared to shell and tube or agitated vessel heat exchangers.

2.2 The Applications Of Thin-Film Processing

Thin-film process equipment is extremely expensive on a heat transfer area basis, costing on the order of 10^4 dollars per square meter for an installed system (Glover, 1987). This necessarily limits their application to materials and products of high value or high value added. In addition, the candidate process must derive real benefits from the characteristics of thin-film equipment, i.e. short residence times and reduced exposure to elevated temperatures, in order to justify the economic selection of this equipment. Pharmaceuticals, cosmetics, biological & biochemical products, as well as many specialty chemicals, meet these requirements. The high, uniform, shear rate applied to the process fluid as it is deposited on the wall by the rotor generally reduces its momentum diffusivity and enhances heat transfer. This 'shear-thinning' effect often makes thin-film processing the only viable alternative for temperature sensitive Bingham plastic and thixotropic materials (Glover, 1987).

2.3 The Current Approach To Design

Thin-film processors are expensive and specialized. Consequently, sales, design, scale up, and manufacturing are significantly integrated. The techniques and objectives of design and scale up are strongly influenced by those of sales and manufacturing. The design chronology normally begins with a customer inquiry. A quantity of the material to be evaluated for processing is forwarded to the vendor's test facility, where it is processed in accordance with a pilot plant testing procedure. Test data are used to calculate the *overall* heat transfer coefficients in the pilot plant processor at several operating points. The heat transfer calculations are based on material and energy balances made at the boundaries of the test unit. Actual calculation of the overall heat transfer coefficients is carried out using the log-mean temperature difference (LMTD) equation.

Scale-up of the process is constrained by the fact that industrial units are manufactured in only eight or ten sizes. These range from 0.5 to 40 square meters of heat transfer surface, with approximate geometric similarity maintained between the sizes. The process is scaled up on the basis of heat transfer area, again using the LMTD equation.

2.4 The Log Mean Temperature Difference Model

The log-mean temperature difference (LMTD) design procedure currently applied to the scale-up of thin-film equipment has been adapted from conventional shell and tube design methods.

The LMTD design equation relates the net heat transferred (Q) to the available area (A), the log-mean temperature difference (ΔT_{LM}), and a characteristic, process specific, constant known as the overall heat transfer coefficient (U). It is U that contains information unique to the particular unit, the physical properties of the process streams, and the processing conditions:

$$Q = UA\Delta T_{LM} \qquad [2.1]$$

Evaluation of the overall heat transfer coefficient is the thermal design objective of the pilot plant experiments. While 'U' can be stated more fundamentally in terms of individual inside and outside *film* heat transfer coefficients (h_I and h_O), this is seldom done in practice. One reason is that heat transfer design data are usually obtained from steady state experiments. There is no way to determine unambiguous values for individual film coefficients from steady state data (this can be done using non-steady state data with statistical techniques, as shown in Chapter 6).

The form of the LMTD equation reflects a specific physical model of process heat transfer. The model is elucidated by deriving expression [2.1]

from a differential energy balance on a simple shell and tube heat exchanger (Kern, 1950). Focusing on Figure 2.3:

$$dQ = U\, dA\Delta T \qquad [2.2]$$

From the slope of the ΔT line,

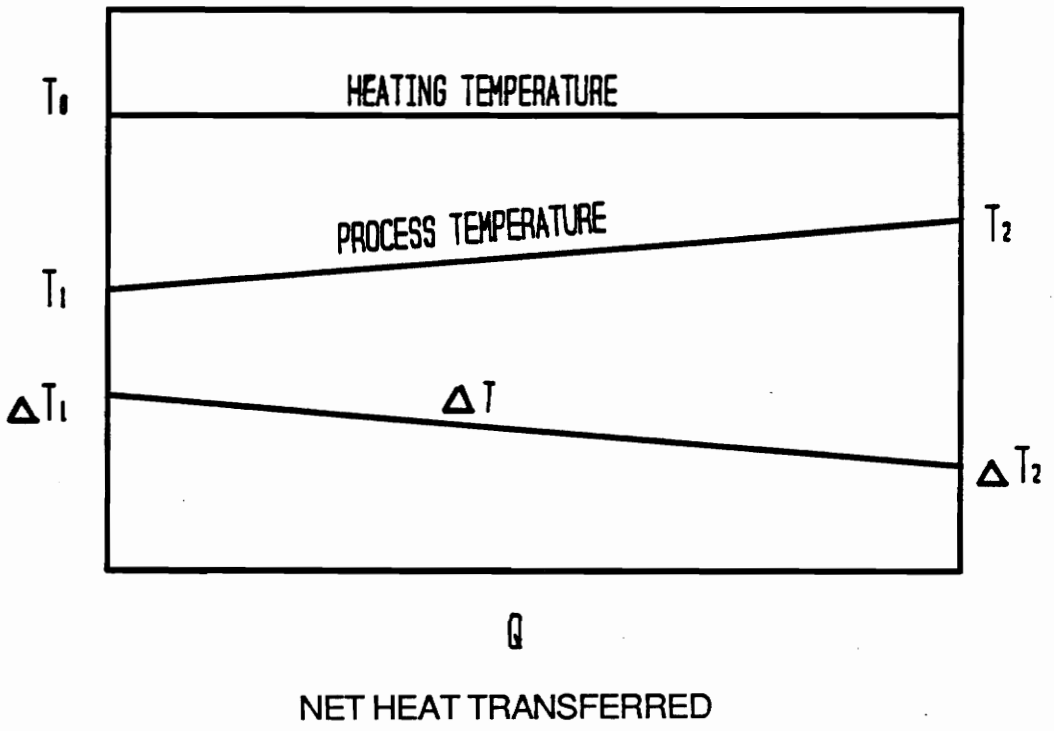
$$\frac{d\Delta T}{dQ} = \frac{\Delta T_2 - \Delta T_1}{Q} \qquad [2.3]$$

Combining [2.2] and [2.3],

$$\int_{\Delta T_1}^{\Delta T_2} \frac{d\Delta T}{\Delta T} = \frac{U}{Q} (\Delta T_2 - \Delta T_1) \int_0^A dA \qquad [2.4]$$

Integrating [2.4],

$$Q = UA \frac{\Delta T_2 - \Delta T_1}{\ln(\Delta T_2/\Delta T_1)} \equiv UA\, \Delta T_{LM} \qquad [2.5]$$



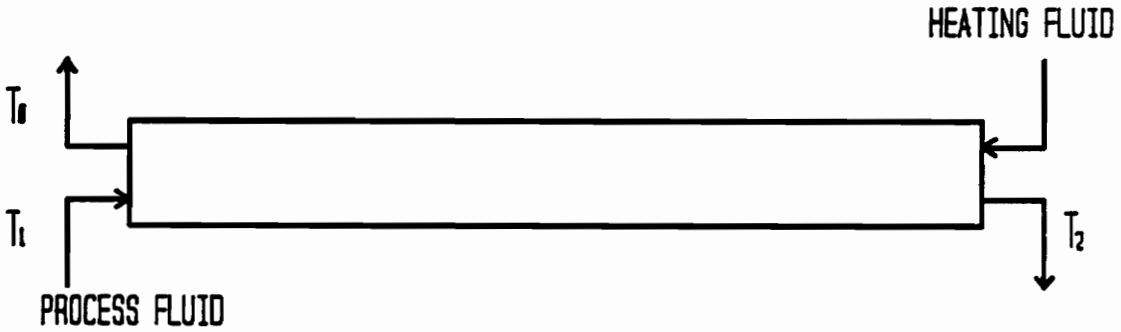


Figure 2.3. The temperature difference diagram for a simple shell and tube heat exchanger with a constant heating temperature T_0 (Kern 1950). Note this model assumes ideal plug flow of the process fluid.

From the derivation it is clear that the log-mean temperature difference applies only to a one-dimensional model of heat transfer. The model permits perpendicular heat transfer between the hot and cold fluids, but there is no mathematical provision for (or physical acknowledgement of) the axial transfer of heat between individual fluid elements in the flowing system. The convection model is one of longitudinally segregated fluid elements with no diffusion or conduction between adjacent elements.

The LMTD model closely approximates reality for heat transfer across a tube wall enclosing fully developed turbulent flow, where the axial velocity profile is flat and radial mixing is nearly complete. The applicability of the LMTD model to heat transfer in thin-film processors is less certain. The action of the rotor and properties of the bow wave might be expected to yield complex axial and radial velocity profiles.

These assertions argue for developing an alternative model of macroscopic fluid convection inside thin-film equipment.

CHAPTER THREE

The Tanks In Series Model

3.1 Lumped Versus Distributed Flow Representations

The search for a simple flow model begins with a choice between representing the system by a lumped parameter model or a distributed parameter model.

The distributed approach involves a direct integration of the differential material balance equations written for the flow system. The simplest distributed model likely to represent the state of convective flow inside a thin-film processor is the one-dimensional dispersion equation (Hopkins et al., 1969):

$$\bar{\tau}\frac{\partial C}{\partial t} + \frac{\partial C}{\partial Z} - \frac{D}{uL}\frac{\partial^2 C}{\partial Z^2} = 0 \qquad [3.1]$$

Where: $\bar{\tau}$ = mean residence time in system
C = concentration (of tracer in tracer studies)
L = characteristic axial dimension
Z = dimensionless axial distance z/L
D = axial dispersion coefficient
u = mean axial velocity $L/\bar{\tau}$
t = time

This equation is also known as the axially dispersed plug flow model. As written, [3.1] represents a non-steady state differential material balance on the tracer at a plane 'z' in the processor. The first term represents

accumulation of the tracer in the differential volume at plane 'z'. The second term represents transport of tracer to plane 'z' by bulk convection. The third term represents the exchange of material between adjacent fluid elements traveling through the system. It is this term that distinguishes the dispersion model from the *segregated* plug flow regime on which the LMTD design equation is based.

The lumped parameter, or state model, approach accommodates axial mixing processes by invoking the concept of a discrete lumped stage, i.e., a finite, localized volume within which all fluid is perfectly mixed. A series, or cascade, of such stages can be used to model flow systems ranging from a well-mixed vessel (one stage) to fully segregated 'plug' flow (infinite stages). The degree of back mixing in the system is represented in terms of the number of stages in the cascade.

It is interesting to note the contrast between the distributed and lumped approaches. The former integrates microscopic gradients to generate a macroscopic model, while the later begins with a macroscopic entity in which gradients are not defined.

Which route is more appropriate for the characterization of macroscopic convection in thin-film processors? Tracer injection experiments can provide sufficient information to estimate the unknown parameters in both models (Ostergaard and Michelsen,1969; Lapidus and Seinfeld, 1974, Chapter 7; Levenspiel, 1972, Chapter 9). Both models can be transformed into the frequency domain in order to expedite the parameter estimation

procedure (Hopkins et al., 1969; Lapidus and Seinfeld, 1974, Chapter 7; Ostergaard and Michelsen, 1969). In terms of providing a flow model, both routes are equally suitable. Does the model's intended end use favor one approach? Yes! The convective flow model is to become the basis for a numerical heat transfer simulation program. The extent to which the model elements can be manipulated as discrete entities materially assists the coding of the simulation. It is on this basis that the lumped parameter modeling approach was selected.

3.2 Flow Model Development

The tanks in series model equations are developed from a non-steady state material balance (the rate equation), for a single component, over the cascade. The system is depicted schematically in Figure 3.1. Here the net mass flow rate is constant while a small, time-varying, mass of tracer is superimposed on the flow. The material balance equations are:

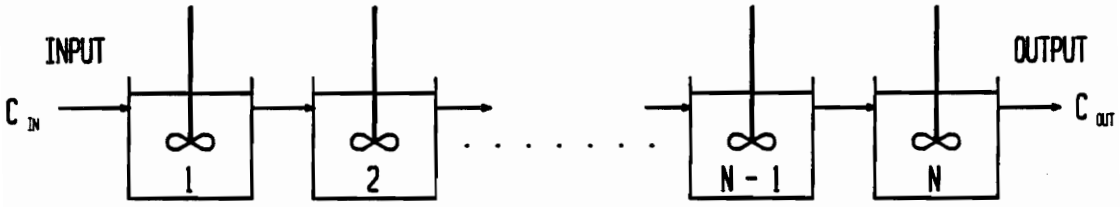


Figure 3.1. A schematic presentation of the ideal tanks in series model. C_{in} and C_{out} represent the inlet and outlet concentrations of an injected tracer.

The overall material balance on the system is:

MASS FLOW RATE IN - MASS FLOW RATE OUT = RATE OF ACCUMULATION

For an incompressible fluid:

$$\rho q_{in} - \rho q_{out} = 0 \qquad [3.2]$$

Hence, $q_{in} = q_{out} = q$

For the tracer in stage 1:

$$qC_{in} - qC_1 = V\frac{dC_1}{dt} \qquad [3.3a]$$

Generalized for the 'i'th stage, [3.3a] becomes:

$$qC_{i-1} - qC_i = V\frac{dC_i}{dt} \qquad [3.3b]$$

Substituting the definition of mean residence time $\tau = V/q$ into [3.3b] yields:

$$\frac{1}{\tau} C_{i-1} - \frac{1}{\tau} C_i = \frac{dC_i}{dt} \qquad [3.4]$$

Beginning with the first stage, and working down the cascade, a system of first-order differential equations can be written:

$$\frac{1}{\tau} C_{in} - \frac{1}{\tau} C_1 = \frac{dC_1}{dt}$$

$$\frac{1}{\tau} C_1 - \frac{1}{\tau} C_2 = \frac{dC_2}{dt} \quad \ldots\ldots \qquad [3.5]$$

The system [3.5] can be represented in matrix notation:

$$\begin{bmatrix} \frac{1}{\tau} & \frac{-1}{\tau} & & & 0 \\ & \frac{1}{\tau} & \frac{-1}{\tau} & & \\ & & & & \\ 0 & & & \frac{1}{\tau} & \frac{-1}{\tau} \end{bmatrix} \times \begin{bmatrix} C_{in} \\ C_1 \\ \vdots \\ \\ C_N \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \\ C_N \end{bmatrix} \qquad [3.6]$$

The input to the system $Y_{in}(t)$ is the inlet concentration:

$$Y_{in}(t) = C_{in}(t) \qquad [3.7]$$

The output of the system $Y_{out}(t)$ is the tracer concentration in the last stage:

$$Y_{out}(t) = C_N(t) \qquad [3.8]$$

Where:
C_{in} = concentration of tracer entering the system
$C_N = C_{out}$ = concentration of tracer leaving the system
C_i = concentration of tracer in stage 'i'
τ = mean residence time in a single stage
q = volumetric flow rate through system
V = volume of a single stage

If the parameters in the leading coefficient matrix, the initial concentrations $C_1(0)...C_N(0)$, and the input $Y_{in}(t)$ are all known, the concentration in each of the stages, or the *state* of the system, can be calculated as a function of time by integrating system [3.6] as an initial-value problem. Any procedure to accomplish this could be termed a

process simulation, albeit a simulation limited to describing macroscopic fluid convection in the system.

Now consider the thin-film processor as a generalized system that is known, or reasonably believed, to behave as a linear cascade of well-mixed stages. In order to characterize fluid convection in the processor it is necessary to evaluate the parameters τ and N. Parameter evaluation using the *time domain* representation of the system (equations [3.6]) is difficult because the dimension of the system (N) is one of the unknown parameters!

Fortunately, when the system of equations is transformed into the Laplace, or frequency, domain, these parameters become more accessible. Rewriting equations [3.5] in the Laplace domain:

$$\frac{1}{\tau}\bar{C}_{in}(s) - \frac{1}{\tau}\bar{C}_1(s) = s\bar{C}_1(s) - C_1(0)$$

$$\frac{1}{\tau}\bar{C}_1(s) - \frac{1}{\tau}\bar{C}_2(s) = s\bar{C}_2(s) - C_2(0) \ldots\ldots. \qquad [3.9]$$

If the initial concentrations in the time domain $C_i(0)$ are zero, rearrangment gives:

$$\frac{\bar{C}_1}{\bar{C}_{in}} = \frac{1}{\tau s+1} \qquad [3.10]$$

$$\frac{\bar{C}_2}{\bar{C}_{in}} = \frac{\bar{C}_1}{\bar{C}_{in}}\frac{\bar{C}_2}{\bar{C}_1} = \frac{1}{\tau s+1}\frac{1}{\tau s+1} \qquad [3.11]$$

$$\frac{\bar{C}_{out}}{\bar{C}_{in}} = \frac{\bar{C}_N}{\bar{C}_{in}} = \frac{1}{(\tau s+1)^N} \qquad [3.12]$$

Where:

s = Laplace domain independent variable
$\bar{C}(s)$ = Laplace transformed tracer conc.
C(0) = time domain initial concentration

Parameters τ and N are now algebraically related to the inlet and outlet concentrations of any observable component, provided the initial concentration of that component in the cascade is zero.

3.3 Estimating Flow Model Parameters

3.3.1 The System Transfer Function

The frequency domain transfer function of a system is defined:

$$G(s) \equiv \frac{\bar{C}_{out}(s)}{\bar{C}_{in}(s)} \qquad [3.13]$$

Equation [3.12] is, by definition, the transfer function for the tanks in series model. Notable is the fact that the transfer function relates all model system parameters to quantities that can be measured at the boundaries of the system (C_{in} and C_{out}). The form of the transfer function suggests that τ and N can be determined by a well known experimental method (see,

for example, Lapidus and Seinfeld, 1974). Suppose an observable tracer compound is added to the inlet stream of the system in a controlled fashion and the tracer concentration in the outlet stream is measured at the same time. These time-dependent concentration data can be transformed into the frequency domain using the discrete Fourier transform (Ramirez, 1985):

$$y(k\Delta f) = \Delta t \sum_{n=o}^{L-1} Y(n\Delta t)\, e^{-j2\pi k\Delta f n\Delta t} \qquad [3.14]$$

Where:

- L = the number of data points
- Δt = the sampling interval
- Δf = the sampling interval in the frequency domain $1/L\Delta t$
- n = the time sample index
- k = the index for the computed set of discrete frequency components

Using the frequency-domain identity $s \equiv j2\pi f$, the Laplace transformed transfer function can be written:

$$G = \frac{\bar{c}_{out}}{\bar{c}_{in}} = (j2\pi f+1)^{-N} \qquad [3.15]$$

3.3.2 The Input Function

Ideally, the input function $C_{in}(t)$ would be made to take a form that transforms into a simple constant in the frequency domain. A suitable choice

is the impulse or delta function. If the impulse is normalized to unity, the frequency domain impulse response $\mathcal{C}_{out}(s)$ becomes:

$$\mathcal{C}_{out} = (1)(j2\pi f\tau+1)^{-N} \qquad [3.16]$$

Equations [3.15] and [3.16] show that, for an impulse experiment, the complete transfer function is determined by measurement of the output variable alone. The restrictions of this method are discussed in Chapter 8.

3.3.3 Parameter Estimation

The original model state equations are now in a form that permits the values of the constant parameters τ and N to be estimated numerically. Any number of non-linear parameter estimation or curve-fitting techniques are applicable. The procedure employed in this work is outlined in the following paragraphs. The computer codes are included in Appendix 6.

The minimum sum-of-squares error (SSE) criteria is widely used to fit non-linear equations to experimental data. This 'goodness-of-fit' criteria works equally well in the time and frequency domain (Clements, 1969). The sum-of-squares error used in this work is expressed in the frequency domain as:

$$SSE = \sum_{k=1}^{L} \left[y_{calc} - y_{obs} \right]^2$$

i.e.,

$$SSE = \sum_{k=1}^{L} \left[y_{calc}(j2\pi f_k) - y_{obs}(j2\pi f_k) \right]^2 \qquad [3.17]$$

Where:

L = number of data points

$y_{calc} = y_{calc}(j2\pi f_k)$ = frequency response predicted by model

$y_{obs} = y_{obs}(j2\pi f_k)$ = normalized system frequency response

3.4 Application To The Thin-Film Processor

A series of tracer impulse experiments was carried out on an industrial pilot plant thin-film processor at the Luwa Corporation test center in Charlotte, North Carolina. The normalized time-domain impulse responses of the processor were recorded over fluid mass flow rates of 50-150 kg/hr and rotor speeds of 5-8.8 m/s. The results are plotted in Appendix 1. The responses were subsequently transformed into the frequency domain using a commercial fast Fourier transform (FFT) code. The points in the transformed data set became the values y_{obs} in the SSE summation. The corresponding values y_{calc} were computed from the transfer function of the model. In succeeding steps, parameter values were adjusted to minimize the sum-of-squares error (SSE).

The SSE equation [3.17] was coded as the objective function in an optimization algorithm (see Appendix 6) designed by Hooke and Jeeves (Kuester and Mize, 1973). The SSE objective function was then minimized with respect to the parameters τ and N. This procedure yielded those values of the stage residence time, and number of stages, that gave the best

model-to-data fit for each set of operating conditions. The operating variables explored in these experiments were the process fluid mass rate and the rotor speed.

The model was extended to permit time shifting, or 'dead time' delay, of the tanks in series response. The resulting transfer function is:

$$G' = e^{-Dj2\pi f} (j2\pi f\tau+1)^{-N} \qquad [3.18]$$

where D is the dead time.

In Appendix 2, the model predictions are plotted against experimental results for both transfer functions [3.15] and [3.18]. The results are summarized in Table 9.1.

The method, described above, used to determine the flow model state equations is summarized, in flowsheet format, in Figure 3.2. The approach is similar to a method developed independently by Calu and Lameloise (1988) for modeling the flow of fluid in climbing-film evaporators.

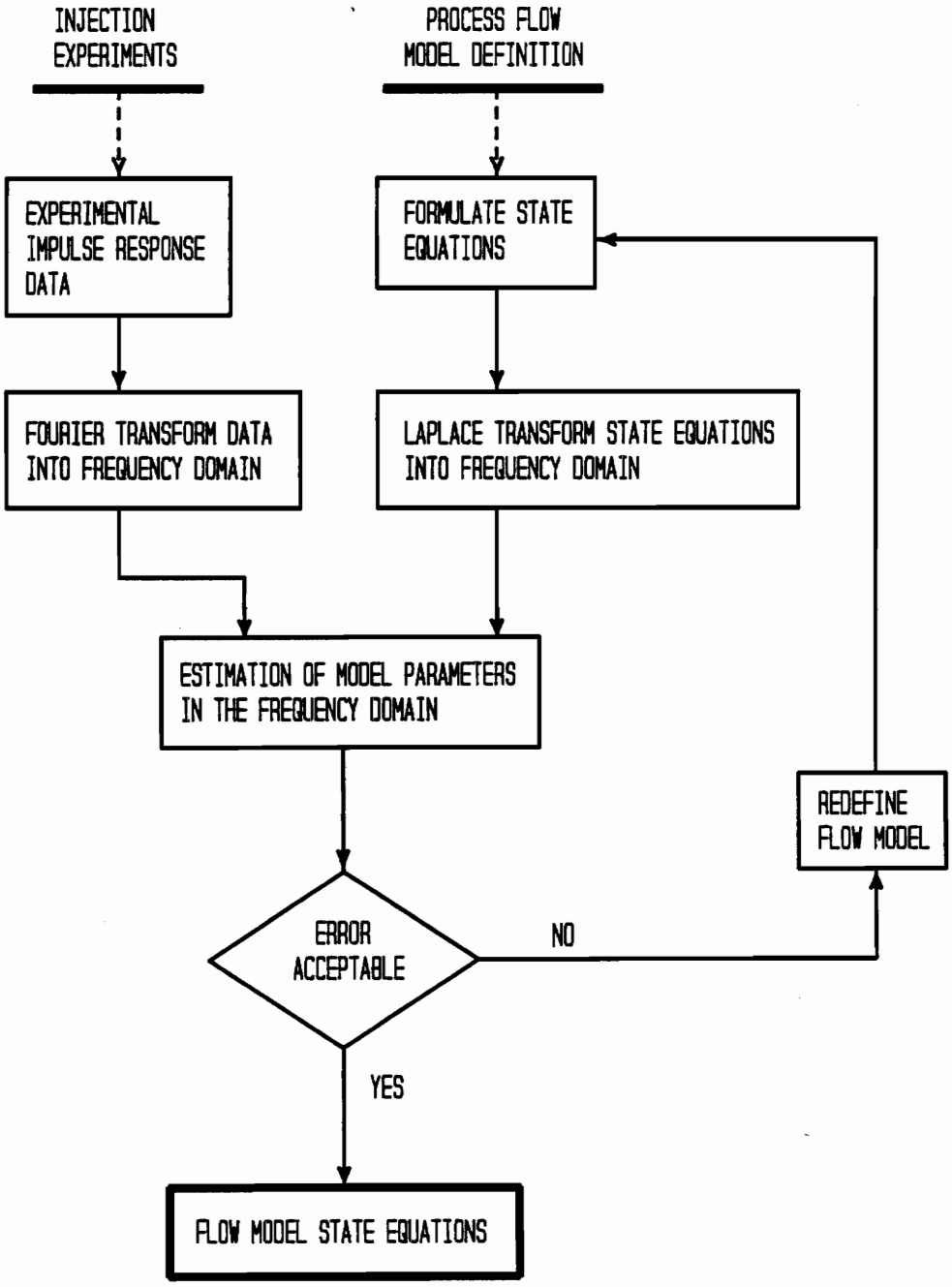


Figure 3.2. The system by which the flow model state equations were tested, refined, and assigned parameter values.

CHAPTER FOUR

Impulse Testing: Experimental Equipment

4.1 The Thin-Film Processor

A general layout drawing of the experimental test unit is provided in Figure 4.1. Detailed drawings of the thin-film processor appear in Figures 4.2 and 4.3. The mean rotor clearance of 1.42 mm was determined by measuring the rotor outside diameter at three equally-spaced points along its axial length and subtracting these values from the inside diameter of the inner shell. The shell diameter was obtained from shop fabrication drawings. It is assumed that the effects of mechanical wear were concentrated in the blades and that wear on the shell was minimal.

The function of the feed distribution ring (Figure 4.2) is to spread liquid evenly around the circumference of the processor before it is taken up by the rotor blades.

The rotor is mechanically stabilized by bearings at the upper and lower ends. Power was provided by a 3 hp.(2.2 kW) three-phase A.C. electric motor and transmitted through a cone-type variable-speed transmission. The available speed range was approximately 1100 to 2100 rpm, which gave a rotor tip velocity ranging from 4.6 to 8.8 meters per second. The rotational speed of the rotor was measured directly using an optical reflectance digital (pulse counting) tachometer.

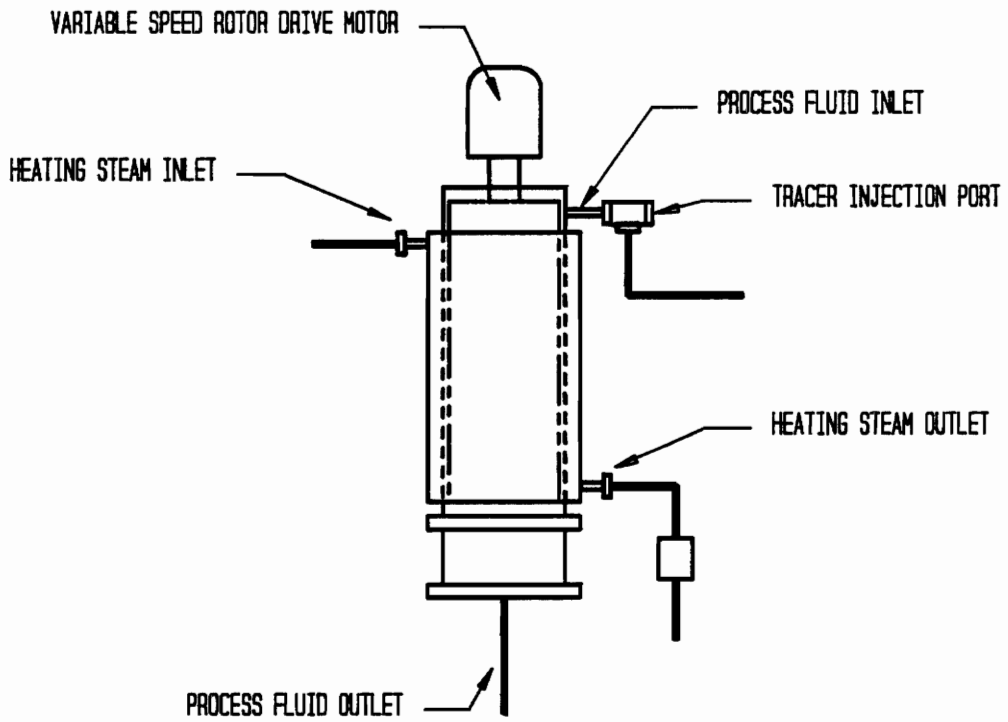


Figure 4.1. A general layout drawing of the thin-film processor used in the thesis experimental program. Note the provisions for steam heating and tracer injection.

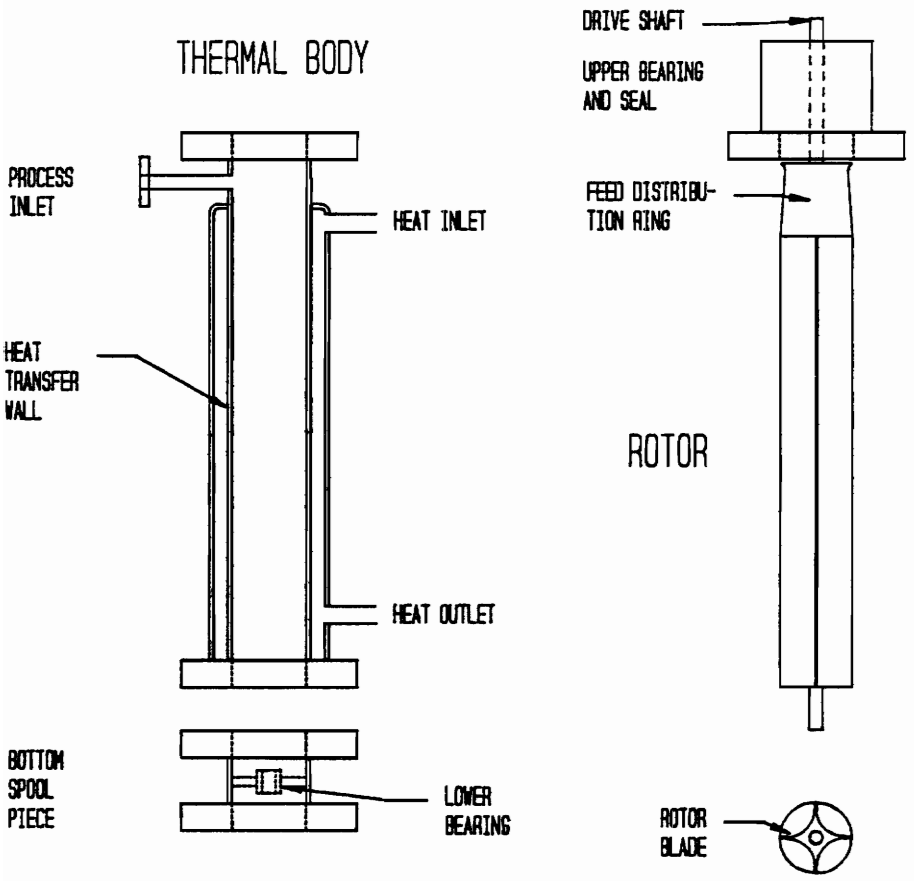


Figure 4.2. A scaled layout drawing of the experimental processor. The upper and lower bearings align the rotor shaft and maintain the rotor clearance. The feed distribution ring spreads entering process fluid uniformly around the heat transfer tube.

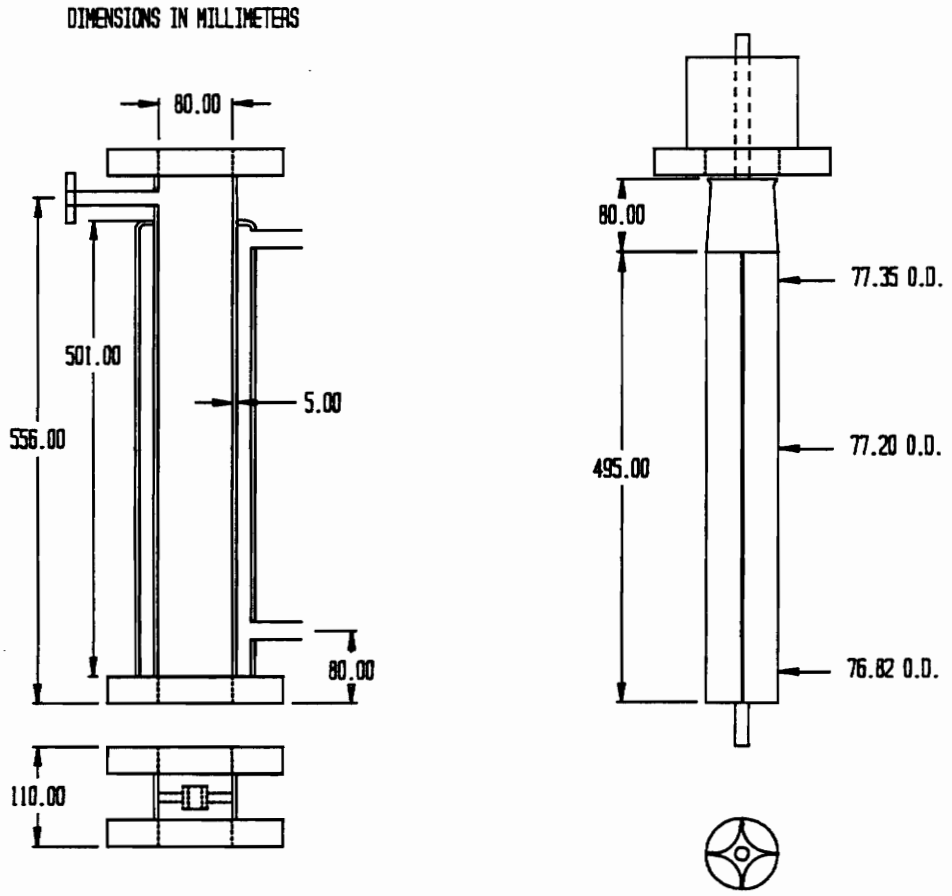


Figure 4.3. The critical dimensions of the experimental thin-film processor shown in millimeters.

4.2 The Feed Control System

Figure 4.4 shows the process control scheme for the fluid feed system. The mass rate of fluid entering the processor was controlled by a conventional pneumatic flow control loop. The flow transducer was a Foxboro Model 13 differential pressure cell with an integral orifice flow element. Orifices ranging in size from 1/32 inch (0.793 mm) to 1/8 inch (3.175 mm) were used. The flow control valve was a 1/2 inch nominal (~15.8 mm I.D.) Badger Research control valve with an equal percentage trim. The controller was a Foxboro Model 43A proportional integral indicating controller. For each orifice, calibration curves were obtained by direct weight measurement, and plotted as mass flow rate versus percent of full scale indicated on the controller. The proportional gain and integral time were adjusted to maximize stability of the system at each operating point. A manually operated ball valve was installed upstream of the processor feed nozzle to act as an on/off block valve for measuring the processor fluid holdup.

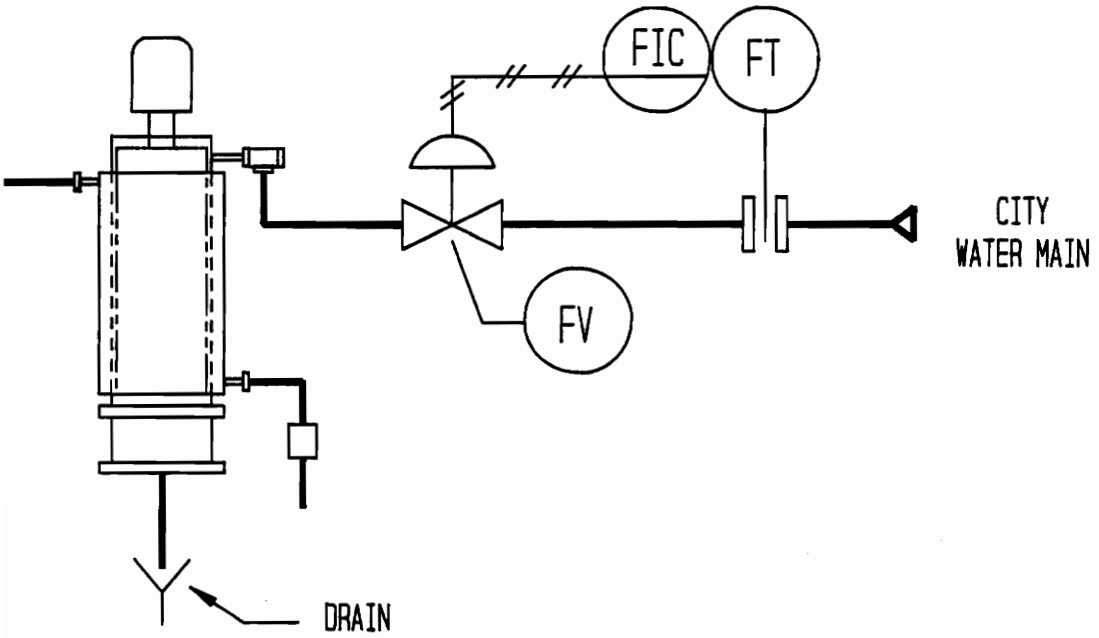


Figure 4.4. The process fluid feed flow rate control system; consisting of a flow transmitter, an indicating flow controller, and a flow control valve. Control loops are drawn to Instrument Society of America (I.S.A.) standards.

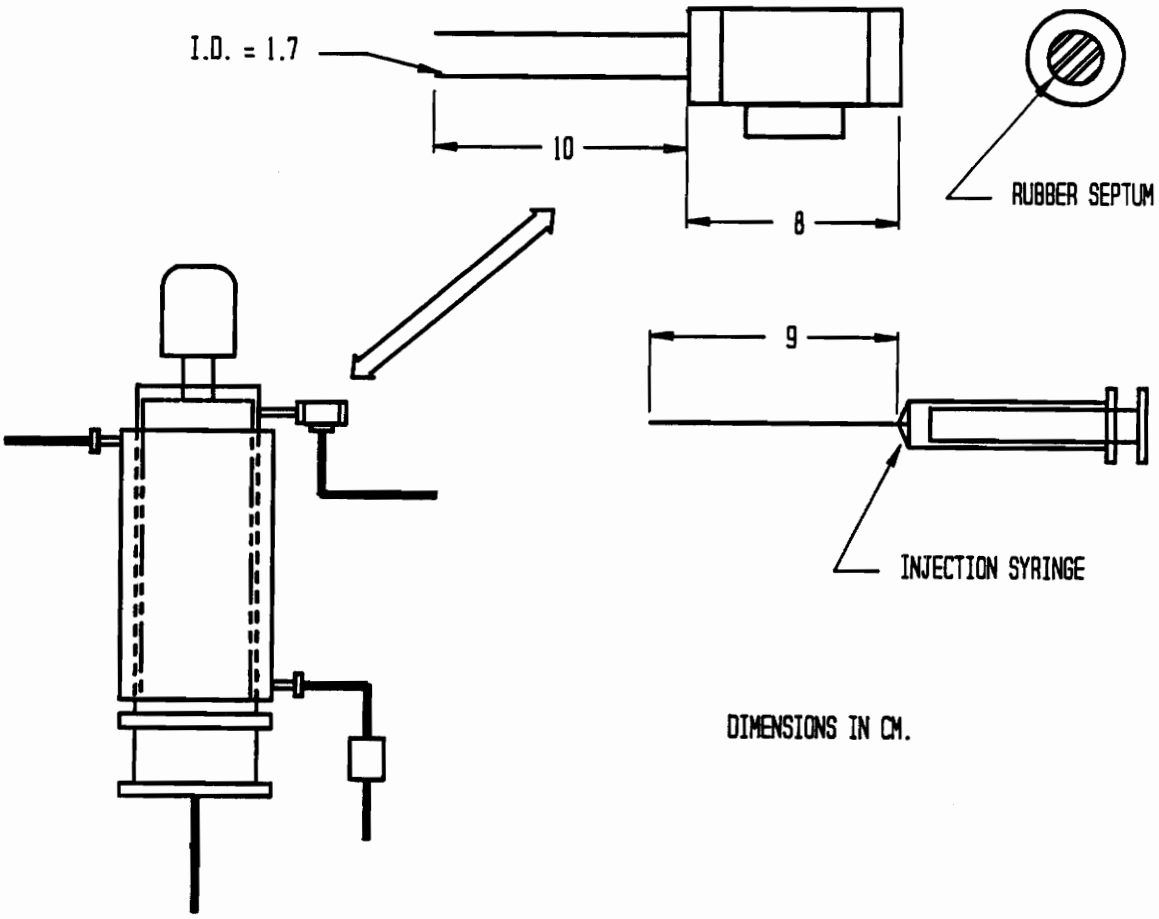


Figure 4.5. Details of the process tracer injection port. Dimensions are in centimeters.

4.3 The Tracer Injection Port

The tracer injection port is shown in schematic form in Figure 4.5. The injection port consisted of a 1/2 inch nominal (~15.8 mm I.D.) tubing tee. The bottom port of the tee was connected to the feed system. The left-hand port was threaded into the thin-film processor feed nozzle flange. The right-hand port of the tee housed a rubberized septum held in place by a 1/2 inch nominal (~15.8 mm I.D.) tubing nut. The tracer was introduced into the processor through the septum using a hypodermic syringe.

4.4 The Conductivity Measurement System

4.4.1 General Description

The tracer used in the experiments was a saturated aqueous potassium chloride salt solution. The tracer concentration in the process was determined from electrical conductivity measurements.

The electrical conductivity of the process fluid (city water) was measured using a self-contained transducer system. The principal system components were the probe, the signal detection electronics, and signal conditioning electronics. The following section contains a brief discussion of the theory of operation.

4.4.2 The Electronics

The electrical conductivity of a dilute aqueous electrolyte solution is proportional to the concentration of ions and to their ionic mobilities (see

Castellan, 1971, Chapter 30). The effect can be detected simply by placing a pair of electrodes in the solution and applying a steady voltage across them. The conductivity is proportional to the ratio of the observed current flow divided by the applied voltage. In practice, it is very difficult to obtain quantitative conductivity data from a steady voltage source. The reason is that as counter-ions accumulate at their respective electrodes, they set up an electric field opposing the applied voltage. The resulting 'capacitor' effect causes the observed current and apparent conductivity to decrease with time. If sufficient voltage is applied, the ions will chemically discharge at the electrode surface and a continuous current will flow. However, the voltage no longer represents the effects of ion mobility, but includes the difference in free energy of the corresponding electrode reactions as well. These difficulties are avoided by:

- limiting the magnitude of the electrode voltage to a level below the discharge potential of the ions in solution
- rapidly reversing the polarity of the electrode voltage

Detection of the resulting electrode current, and conditioning of the signal into a desired output format were the basic design requirements for the conductivity measurement system shown in block diagram form in Figure 4.6.

The system is divided into five blocks. The conductivity element

amplifier, precision rectifier, and signal integrator modules were adopted from Khang and Fitzgerald (1975) with minor modifications. The conductivity element functions as a variable-feedback impedance in the conductivity element amplifier. The bandpass filter module was added to insure that the signal would not be corrupted by the a.c. line frequency of 60 Hz or by stray radio frequency transmissions present in an industrial setting. The output amplifier served to provide the correct polarity, level, and impedance for the output signal. A detailed description of the system electronics is included in Appendix 5.

4.4.3 The Probe

The actual conductivity element (CE) was a rugged, stainless steel probe fabricated from a straight section of T-304 stainless steel tubing, enclosing a length of stainless steel welding rod, as shown in Figure 4.7. Prior to assembly, the outside diameter of the rod was increased by applying successive layers of heat-shrinkable insulation to the rod. Sufficient insulation was applied to create a tight sliding fit between the rod and the inside diameter of the tubing. The insulation served both to center the rod inside of the tubing and to electrically isolate the rod from the tubing. The probe was assembled by inserting the rod into the tubing, with one end flush, and allowing 5 cm of rod to extend from the other end of the tube.

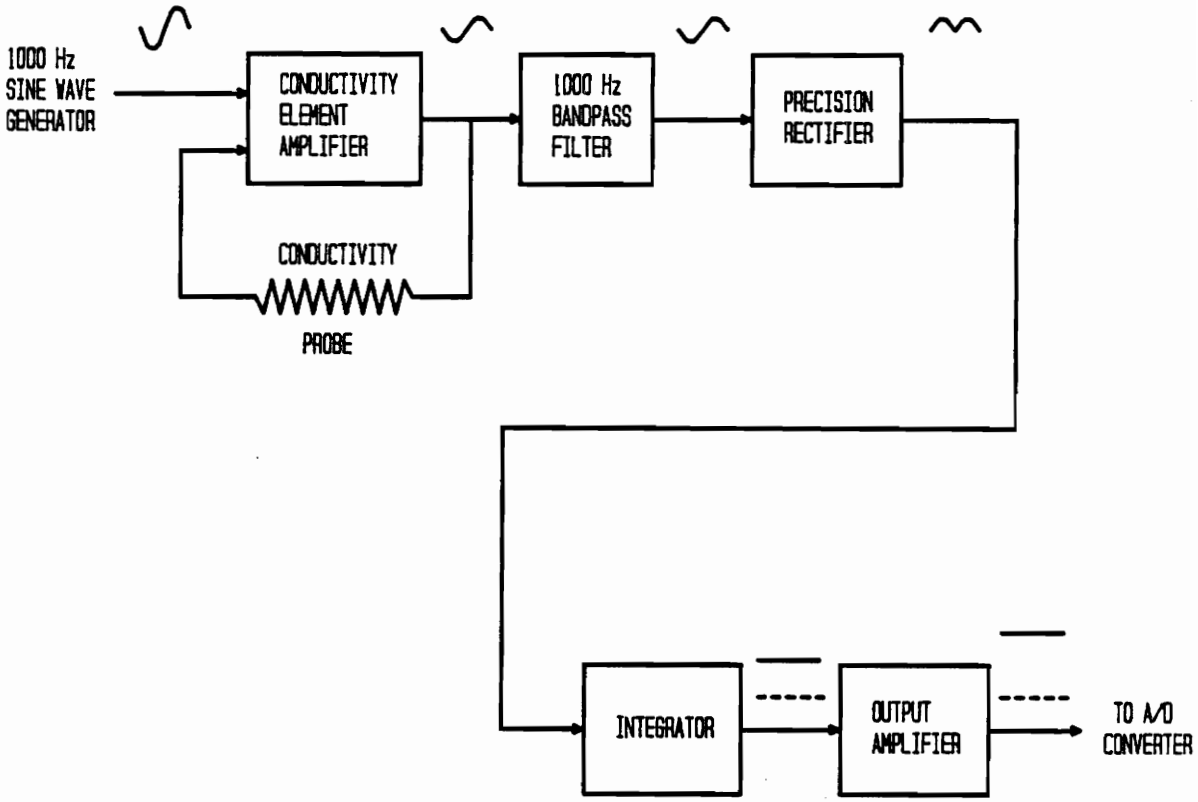


Figure 4.6. A block schematic representation of the process conductivity measuring system.

Both ends were sealed with epoxy and cured. The flush end was filed flat, then wet sanded with 240, 400, and 600 grit paper. It was finally polished with Brasso (R) brass polish to a highly reflective finish. This became the conductivity sensing end; the tip of the rod was the A.C. signal electrode, and the tip of the tube was the ground electrode. The opposite end of the probe was connected to the external circuit: the rod extending from this end was trimmed to the male core dimensions of a BNC type cable connector and a connector was subsequently fitted. Details of the probe assembly are shown in Figure 4.6.

In the early stages of the experimental program the probe would occasionally cycle between a wetted and a non-wetted state. This resulted in signal 'dropout' (i.e., total discontinuation of the signal) and excessive noise. The problem could not be rectified simply by immersing the probe in a collecting cup at the discharge of the unit because the dynamic response of the cup would then be recorded along with that of the unit. An acceptable solution was arrived at by the attachment of a small (less than 1 ml.) reservoir to the end of the probe in the form of a split piece of tygon tubing (Figure 4.8).

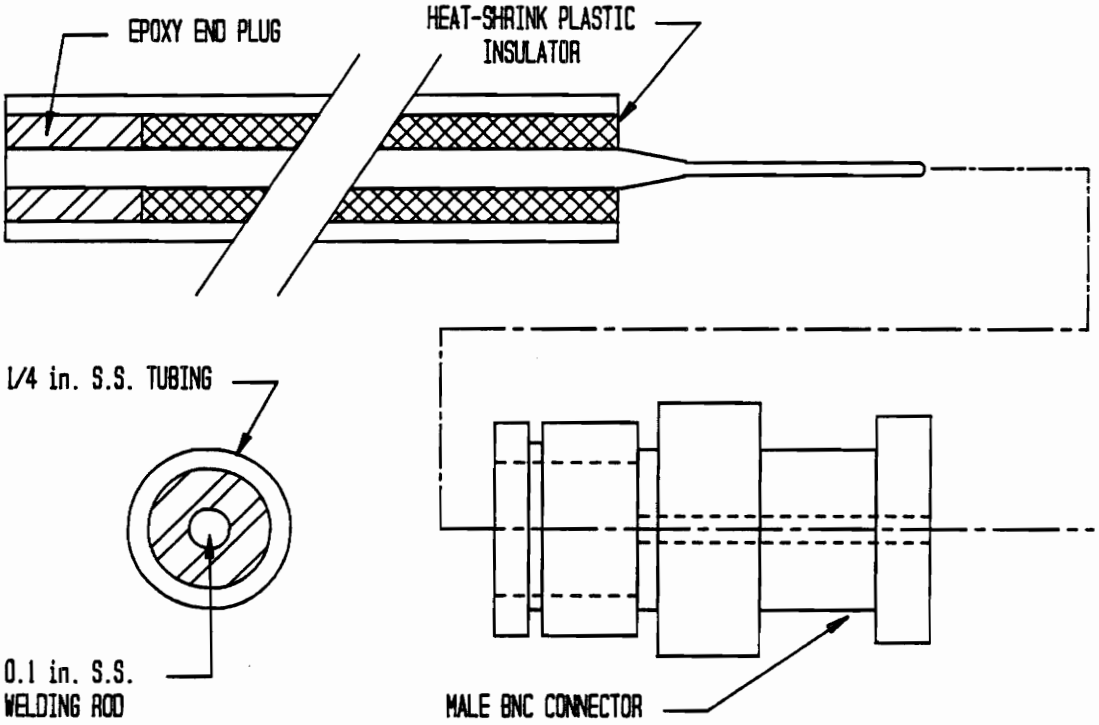


Figure 4.7. Construction details of the conductivity probe. The only electro-active surfaces are the ends of the stainless steel tubing and the centered rod. The tubing outside wall is insulated with electrical tape (not shown).

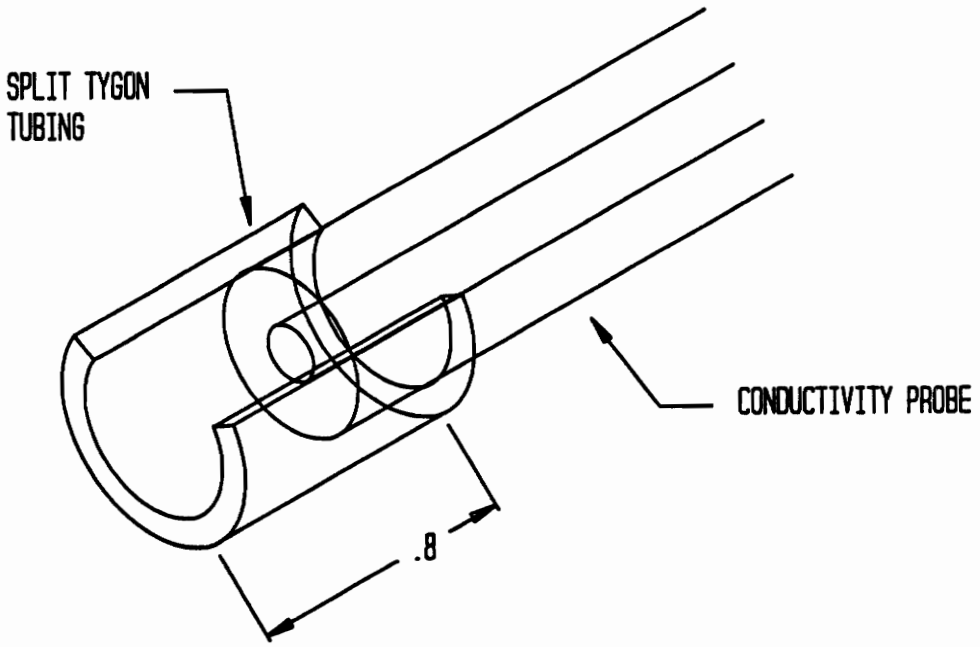


Figure 4.8. The micro reservoir pictured was placed on the tip of the conductivity probe to insure proper electrode wetting during the tracer experiments. The reservoir was kept deliberately small to minimize the response time of the probe (dimensions in centimeters).

CHAPTER FIVE

Impulse Testing: Experimental Procedures

5.1 Tracer Injection

A saturated aqueous potassium chloride solution was used as the tracer material. The tracer injection technique was designed to simulate a finite impulse input.

An acceptably narrow pulse was achieved by using a small injection volume of 0.20 ± 0.02 ml. The pulse was delivered with a hand-held hypodermic syringe through the rubber septum in the injection port shown in Figure 4.4. Synchronization of the injection to the data logging time base was accomplished manually. In order to prevent a 'leading signal', caused by the diffusion of tracer out of the needle before the plunger is depressed, about 0.01 ml. of air was drawn into the syringe. The needle was then wiped dry and inserted to its full length in the injection port.

5.2 A Protocol For The Impulse Response Experiments

1. Set the rotor speed and feed rate at the desired values for the test.
2. Clamp the conductivity probe to the processor bottom spool piece in such a way that its tip intercepts one of several rivulets of fluid discharging from the circumference of the

spool piece.

3. Adjust the conductivity transducer baseline for an output of zero.
4. Initialize the data logging program setup file and record the run parameters.
5. Prepare the syringe and insert it into the injection port.
6. Rapidly, and simultaneously, depress the syringe plunger and the space bar on the computer keyboard to begin logging data.

5.3 Conductivity Transducer Calibration

The conductivity transducer was calibrated using Charlotte, N.C. city water as a standard or zero reference solution. This selection was necessary because of the high inherent conductivity of the city water which would otherwise introduce a significant bias into the experimental data. A set of calibration standards was created by adding specified weights of potassium chloride (KCl) to specified weights of city water according to the relation:

$$\text{Wt. percent KCl} = [\text{Wt. KCl}/(\text{Wt. KCl}+\text{Wt. } H_2O)]\times 100 \qquad [5.1]$$

The standards are represented by the calibration points on the conductivity transducer calibration curve in Figure 5.1.

The slope of the conductivity versus concentration curve is

theoretically constant for low salt concentrations (Khang and Fitzgerald, 1975). The range of concentrations used in the experimental program extends well beyond the region of constant slope. Hence, a polynomial curve fit is needed of the calibration data to provide a continuous analytic expression for converting experimental transducer voltage to true weight percent KCl tracer concentration. The rationale for using these high concentrations involves a tradeoff. The desirable linear portion of the curve includes tracer conductivities on the same order of magnitude as the conductivity of the process fluid (city water). The resulting noise-to-signal ratio is large, as is the influence of variations in process fluid conductivity on the signal. The conductivity transducer response is adjustable in baseline, and magnitude, by 'zero' and 'gain' controls. The circuit details are presented in Appendix 5.

5.4 Preparation of the Conductivity Probe

The polished electrode surfaces at the probe tip were susceptible to chemical reaction and physical contamination from normal handling. A common procedure for passivating electrode surfaces, i.e. immunizing them against undesired chemical interactions, is to apply a thin surface coating of 'black platinum' electrolytically. The platinized surface is resistant to oxidation and other surface reactions that can cause its electrical properties to change over time.

Although platinization is the optimal electrode preparation from a

chemical standpoint, there were concerns about the mechanical stability of the platinum film in an abusive pilot plant environment as well as its resistance to harsh cleaning procedures required to remove grease and dirt. This prompted development of an alternative procedure employing a jellied automotive rust inhibitor manufactured by the Radiator Specialty Co.. The active component is phosphoric acid.

The probe was prepared for an experimental run by immersing the tip in the jellied phosphoric acid for approximately one minute, then vigorously rinsing away the gel. This was followed by immersion in running water for two minutes. During this immersion time, the output of the conductivity transducer rapidly decreased to a steady value and was stable for about 30 minutes. The procedure was repeated at 30-minute intervals.

5.5 Sources Of Experimental Error

5.5.1 Conductivity Calibration Curve

The polynomial calibration curve was fit to calibration data points by a non-linear least squares curve fitting program rather than a more appropriate linear regression method. The polynomial coefficients do not generate an optimal fit of the calibration curve to the data. The calibration curve error causes some amplification of the concentration signal at high voltage and attenuation of the signal at low voltage. The net result is a measured response that is somewhat steeper and narrower than the true response. The range of the response signal normally fell between zero and three volts. The maximum concentration error in this range is less than 15 percent of the 3 volt value.

5.5.2 The Processor Bottom Spool Piece

The rotor in the processor used for the experiments was stabilized by a bearing housed in a spool piece located below the heating jacket as shown in Figure 4.2. The conductivity measurements were made below the spool piece. Hence, its area was incorporated into the flow model. This is problematical because the area contributes to flow dispersion but not to heat transfer.

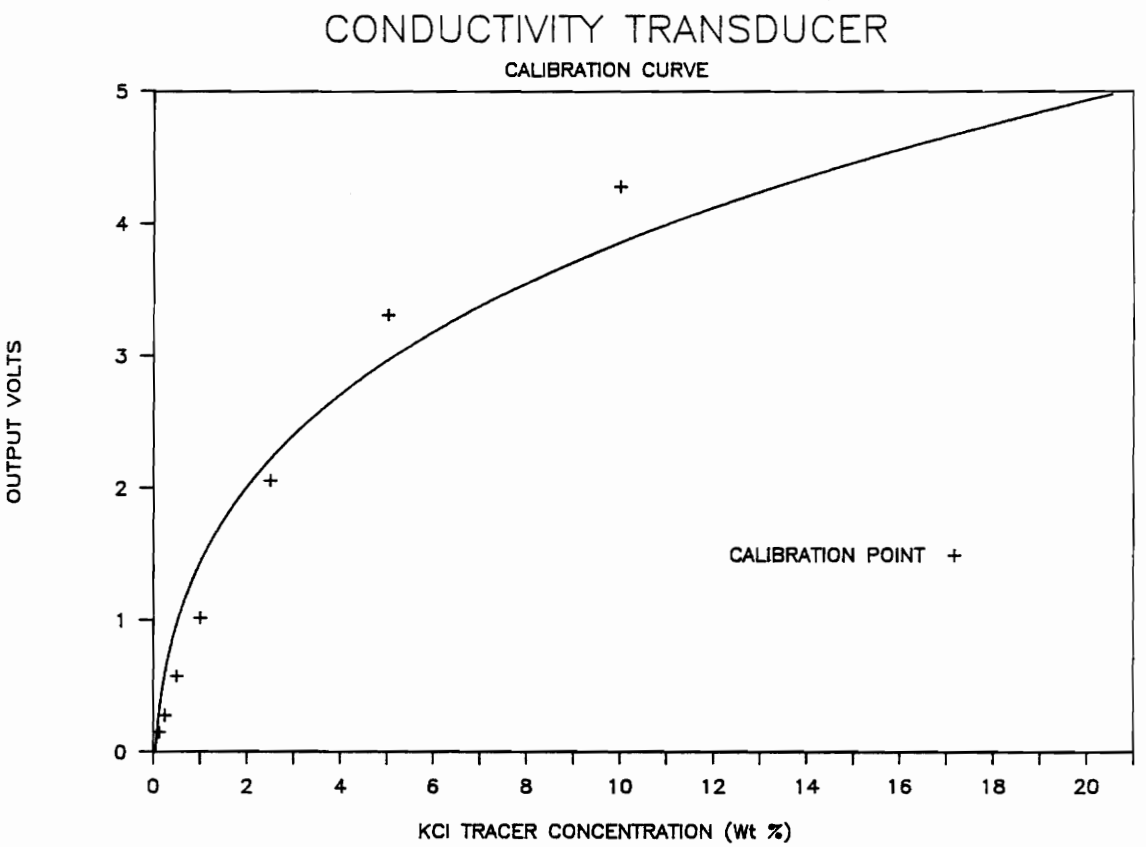


Figure 5.1. The conductivity transducer calibration curve. Points (+) indicate experimentally determined values. The solid curve represents the calibration equation used to convert transducer voltage to tracer concentration. The discrepancy between predicted and experimental values is addressed in section 5.5.1.

5.5.3 Location Of the Injection Port

The fact that the tracer injection did not occur exactly in the plane delineating the start of the heat transfer surface introduces uncertainty in the measured dead times.

5.5.4 Electrode Reactions

Chemical reactions at the electrode surfaces continuously passivated the conductivity element. The effect of passivation manifested itself primarily in a slow drift in transducer sensitivity. The effect was largely offset by normalizing the experimental responses.

CHAPTER SIX

The Processor Heat Transfer Simulation

6.1 The Purpose And Approach

The motive for writing and running a detailed heat transfer simulation of the thin-film processor was two-fold. First, to ascertain whether a tanks in series model for fluid convection could be incorporated into a numerical heat transfer simulation of the thin-film processor. Second, to make a quantitative evaluation of the intrinsic or dispersion corrected heat transfer film coefficients proposed in chapter one.

Evaluating the intrinsic film coefficients involved three areas of activity:

1. Running a series of transient heat transfer experiments using the thin-film processor pilot plant. During these experiments the heating steam and process fluid discharge temperatures were recorded. The experimental details are described in chapter 7.

2. Coding of the numerical simulation. This involved selecting and interfacing commercial codes for the numerical solution of initial and boundary-value problems. In addition, it was necessary to write special code to interface these programs to the time-varying boundary conditions of the transient simulation.

3. Values for the intrinsic film coefficients were obtained by

systematically selecting those values which, when substituted into the simulation, gave the best fit between the simulated temperature response and the corresponding experimental response. Coefficients obtained in this way *do not contain the effects of dispersion or back mixing*.

6.2 Heat Transfer Model Formulation

6.2.1 Building On The Tanks In Series Model

The tanks in series model is the basis upon which the heat transfer model, and subsequent simulation, were written. Each tank in the cascade is treated as an independent heat transfer stage.

Another conservation relationship was then invoked, this time energy conservation. However, for a given heat transfer stage, non-steady state energy balances must be written around *two* separate entities: the process fluid contained within the stage boundaries, *and* the segment of heat transfer wall associated with that stage.

The energy balance on the fluid within each stage is handled by a lumped capacitance temperature model. This is in keeping with the stirred-tank representation of fluid mixing in the processor. The unsteady energy balance on the processor heat transfer wall, however, returns to the problem of a lumped versus a distributed model.

6.2.2 The Lumped Wall Model

Since lumped models do not allow for gradients in intensive variables such as temperature, the lumped model requires the temperature to remain uniform throughout the wall. This allows both the wall temperature and the fluid temperature to be stated as a function of a single variable, time. The advantage of this modeling approach is the ready transformation of the resulting expressions into a system of linear ordinary differential equations (O.D.E.'s). The temperature-time trajectory of such a system can be easily calculated using standard O.D.E. solving methods.

The lumped model system (Figure 6.1) equations are derived below beginning with an energy balance on the process fluid. There are three pathways for energy transport into the process fluid:

1. Thermal energy convected into the stage with the process fluid as sensible heat.
2. Thermal energy convected from the surface of the heat transfer wall.
3. Mechanical energy imparted directly to the fluid by the action of the rotor.

The energy balance on the fluid takes the form:

ACCUMULATION = CONVECTION FROM FLUID

+ CONVECTION FROM WALL

+ GENERATION [6.1]

In terms of process variables for stage 'i':

$$P_i + Q_{fi} + wCpT_{i-1} - wCpT_i = M_f C_p \frac{dT_i}{dt} \quad [6.2]$$

Where:

P = power input from rotor Energy/t
$Q_f = Q_{wall\ out}$ = energy convected from wall
w = process fluid mass flow rate
C_p = process fluid heat capacity
T = process fluid temperature
M_f = mass of fluid in the heat transfer stage

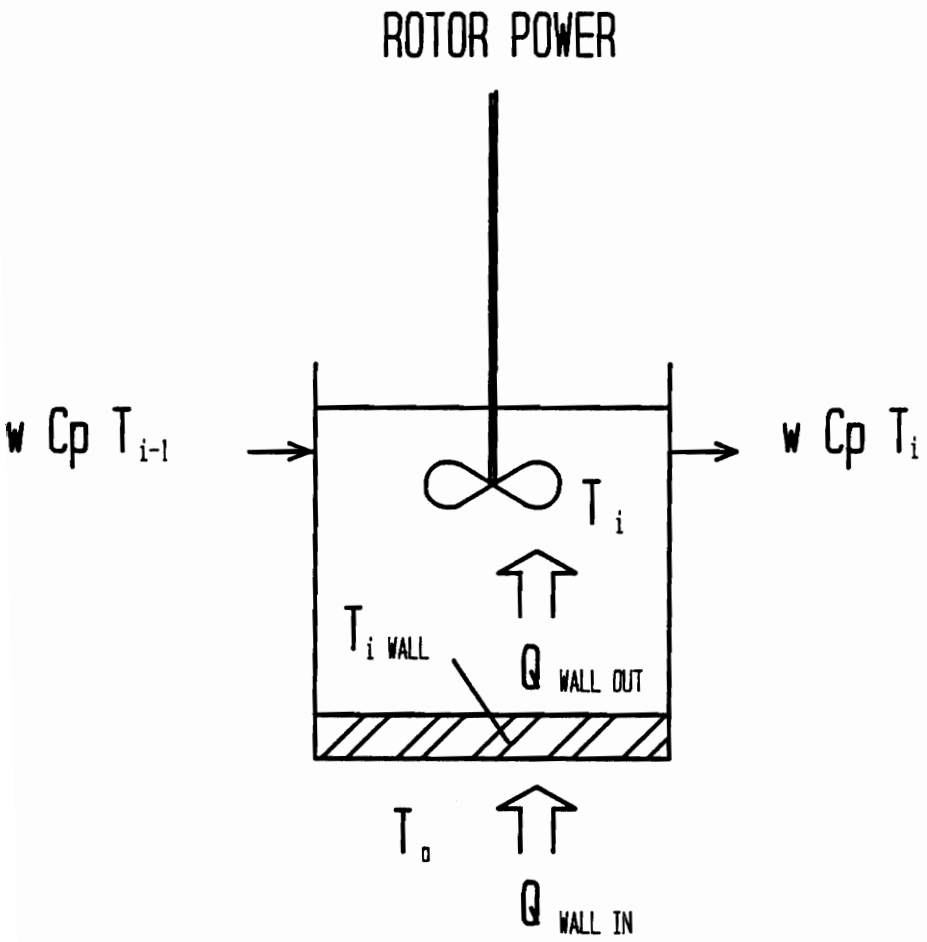


Figure 6.1. An idealized heat transfer stage with a lumped capacitance heating wall. This model augments an ideal stirred tank in fig. 3.1 with an energy balance which includes heat transferred across the processor wall and mechanical energy input by the rotor. The lumped wall is modeled with a uniform internal temperature T_{wall}. The absence of a wall temperature gradient produced an unacceptable error in the subsequent heat transfer simulation.

There are two pathways for energy transport into the heat transfer wall:

1. Thermal energy convected into the wall from steam on the heating side.
2. Thermal energy convected out of the wall by fluid on the process side.

The corresponding energy balance on the heat transfer wall is:

CONVECTION FROM HEATING STEAM

+ CONVECTION FROM FLUID

= ENERGY ACCUMULATION IN WALL [6.3]

$$Q_{wi} - Q_{fi} = M_w C_w \frac{dT_{wi}}{dt} \quad [6.4]$$

with initial conditions at t=0,

$$T = T_w = T_{in} \quad ; \quad \frac{dT}{dt} = \frac{dT_w}{dt} = 0 \quad [6.5]$$

and boundary conditions at the inside and outside surface of the heat transfer wall,

$$Q_{fi} = h_I (T_w - T)_i \quad ; \quad Q_{wi} = h_O (T_O - T_w)_i \quad [6.6]$$

where:

M_w= mass of the heat transfer wall associated with each stage
C_w= heat capacity of the wall
Q_w= energy convected into wall on steam side
Q_f= energy convected out of wall on process side
T_w= wall temperature
T_{in}= initial processor inlet temperature
h_I= process side (inside) film heat transfer coefficient Energy/(t deg.)
h_O= heating side (outside) film heat transfer coefficient Energy/(t deg.)
T_O= heating steam temperature

Note: film coefficients are *specific*, i.e., evaluated per unit area

The model equations for all of the individual stages can be written together as a linear system.

For the fluid:

$$\frac{P_i}{M_f C_p} + \frac{h_I}{M_f C_p} T_{wi} - \frac{h_I}{M_f C_p} T_i - \frac{1}{\tau} T_i + \frac{1}{\tau} T_{i-1} = \frac{dT_i}{dt} \qquad [6.7]$$

For the wall:

$$\frac{h_O}{M_w C_w} T_O - \frac{h_O}{M_w C_w} T_{wi} - \frac{h_I}{M_w C_w} T_{wi} + \frac{h_I}{M_w C_w} T_i = \frac{dT_{wi}}{dt} \qquad [6.8]$$

The systems [6.7] and [6.8] are expressed in matrix notation as:

$$\begin{bmatrix} A & B & \dots & 0 \\ & A & B & \\ 0 & \dots & A & B \end{bmatrix} \times \begin{bmatrix} T_{in} \\ T_1 \\ \vdots \\ \\ T_N \end{bmatrix} + C \begin{bmatrix} T_{w1} \\ T_{w2} \\ \vdots \\ \\ T_{wN} \end{bmatrix} + D \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \\ 1 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ \\ T_N \end{bmatrix} \qquad [6.9]$$

$$E \begin{bmatrix} T_{in} \\ T_1 \\ \vdots \\ \vdots \\ \\ T_N \end{bmatrix} - K \begin{bmatrix} T_{w1} \\ T_{w2} \\ \vdots \\ \\ T_{wN} \end{bmatrix} + F \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \\ 1 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} T_{w1} \\ T_{w2} \\ \vdots \\ \\ T_{wN} \end{bmatrix} \qquad [6.10]$$

$$A = \frac{1}{\tau} \; ; \quad B = - \frac{h_I}{M_f C_p} - \frac{1}{\tau} \; ; \quad C = \frac{h_I}{M_f C_p} \; ; \quad D = \frac{P}{M_f C_p} \qquad [6.11]$$

A, B, and C are the constant factors in the terms representing energy input to the fluid by the combined effects of thermal and fluid convection.

D is the mechanical energy input by the rotor.

$$E = \frac{h_I}{M_w C_w} \; ; \quad F = \frac{h_O}{M_w C_w} T_O \; ; \quad K = \frac{h_I}{M_w C_w} + \frac{h_O}{M_w C_w} \qquad [6.12]$$

E, F, and K are constant factors for the terms representing the energy input to the *wall* by thermal convection.

A simulation program based on the above system of energy balance equations, and the material balance equation [3.1], was written using the International Mathematics and Statistics Library (Houston, Texas)

initial-value problem solving code DVERK. The program was coded as a feasibility demonstration of the tanks in series concept for a heat transfer simulation of the thin-film processor. The input function was coded as an ideal step input in heating temperature T_O. The heat transfer film coefficients were assigned typical industrial values of h_O = 8500, and h_I = 500 W m^{-2} K^{-1} (Glover, 1987).

While the results of the simulation were encouraging, the lumped parameter assumption of quasi-static equilibrium within the thermal wall can be shown to be a poor approximation of the actual thermal state of the wall during operation.

For example, during a typical experiment, the energy flux through the wall of the pilot plant processor was 4×10^4 W m^{-2}. For a wall conductivity of 16.3 W m^{-1} K^{-1}, and a wall thickness of 0.005 m, the temperature drop across the wall is given by:

$$\frac{Q}{kA}\,\Delta x = \Delta T \quad ; \quad \frac{40000}{16.3 \times 1} = 12.3 \text{ deg. C} \qquad [6.13]$$

A temperature drop of this magnitude cannot be neglected. This is especially true during the critical first seconds of the processor's transient response, when temperature gradients in the stage walls are developing.

6.2.3 The Distributed Wall Model

Because internal temperature gradients in the heat transfer wall cannot be ignored, the lumped parameter characterization of the wall must be replaced with the distributed model in Figure 6.2. The energy balance now becomes a differential-difference equation:

$$Q_x - Q_{x+\Delta x} = \rho C_w \frac{\partial T_w}{\partial t} \Delta x \qquad [6.14]$$

and

$$Q_x = -k \frac{\partial T_w}{\partial x}\bigg|_x \; ; \quad Q_{x+\Delta x} = -k \frac{\partial T_w}{\partial x}\bigg|_x + \left[\frac{\partial}{\partial x}\left[-k \frac{\partial T_w}{\partial x} \right] \right]\bigg|_x \Delta x \qquad [6.15]$$

Substituting for Q_x and in equation [6.14] using equation [6.15], then dividing through by Δx, yields a partial differential equation with the form of the classic heat conduction equation:

$$k \frac{\partial^2 T_w}{\partial x^2} = \rho C_w \frac{\partial T_w}{\partial t} \qquad [6.16]$$

with boundary conditions

$$-k \frac{\partial T_w}{\partial x}\bigg|_{x=o} = h_O \left[T_O - T_w\big|_{x=o} \right] \; ;$$

$$-k \frac{\partial T_w}{\partial x}\bigg|_{x=L} = h_I \left[T_w\big|_{x=L} - T \right] \qquad [6.17]$$

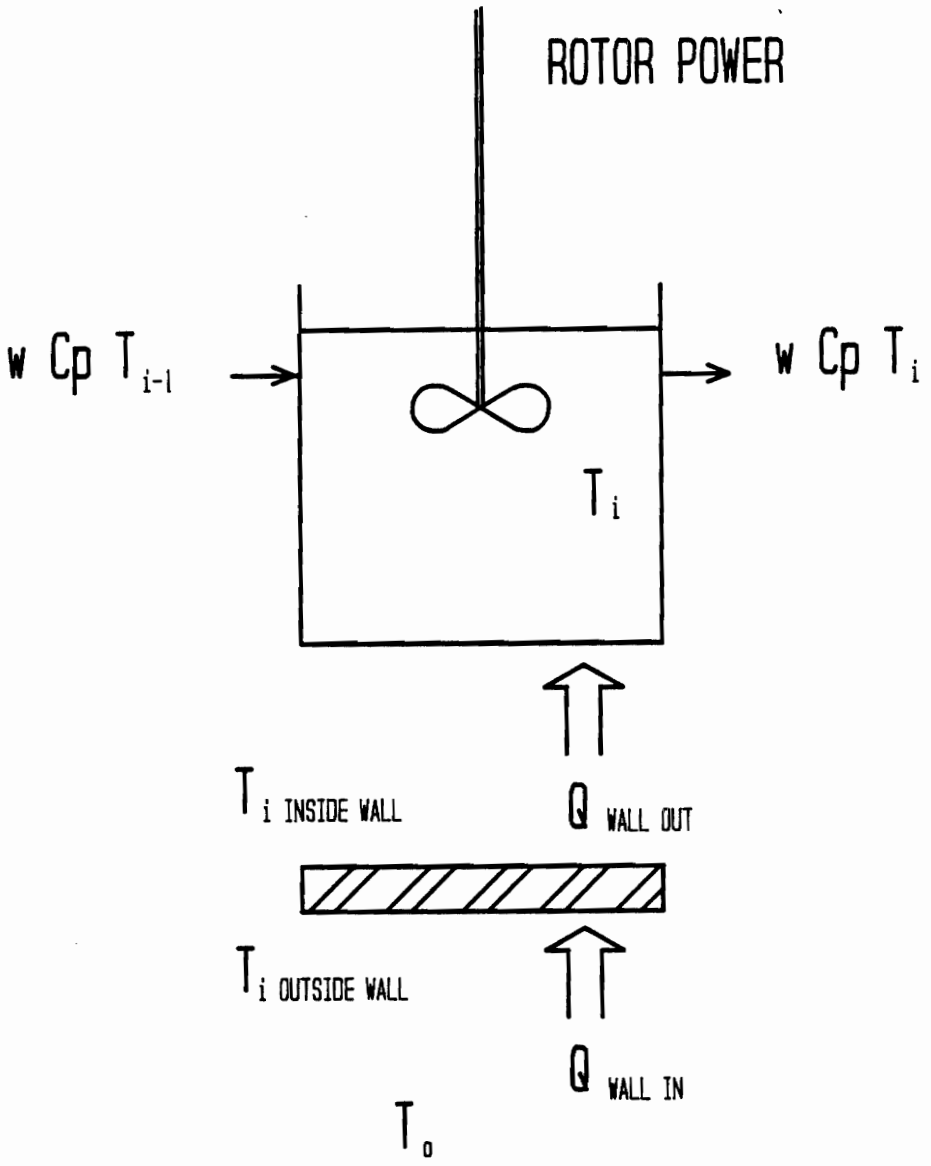


Figure 6.2. The idealized model stage with a distributed heat transfer wall. The inside and outside wall temperatures are determined by the internal wall temperature gradient.

6.3 Elements And Structure Of The Simulation

6.3.1 Overview

A distributed model of the heat transfer wall permits internal gradients of intensive properties. However, the expression and solution of the system equations becomes more complicated. The energy balance equations on the process fluid remain in the format of an initial-value problem, but the system of energy balance equations on the wall elements now constitutes a boundary-value problem (B.V.P.). Boundary-value problems have at least two independent variables, in this case position 'x', and time 't'. While the partial differential equations remain linear, O.D.E. solving codes such as DVERK are not applicable to them. Instead, a class of finite element codes using an approach known as the 'method of lines' has been developed expressly for the solution of this type of B.V.P. (see, for example, the codes referenced and described by Davis, 1984). The IMSL subroutine DPDES is such a program and was used in the final implementation of the processor heat transfer simulation.

6.3.2 Logic And Task Organization

Figure 6.3 shows the logical construction of the process simulation program. The tasks required to evaluate the temperature and flux in the wall are paralleled by those needed to determine the temperature in the bulk fluid

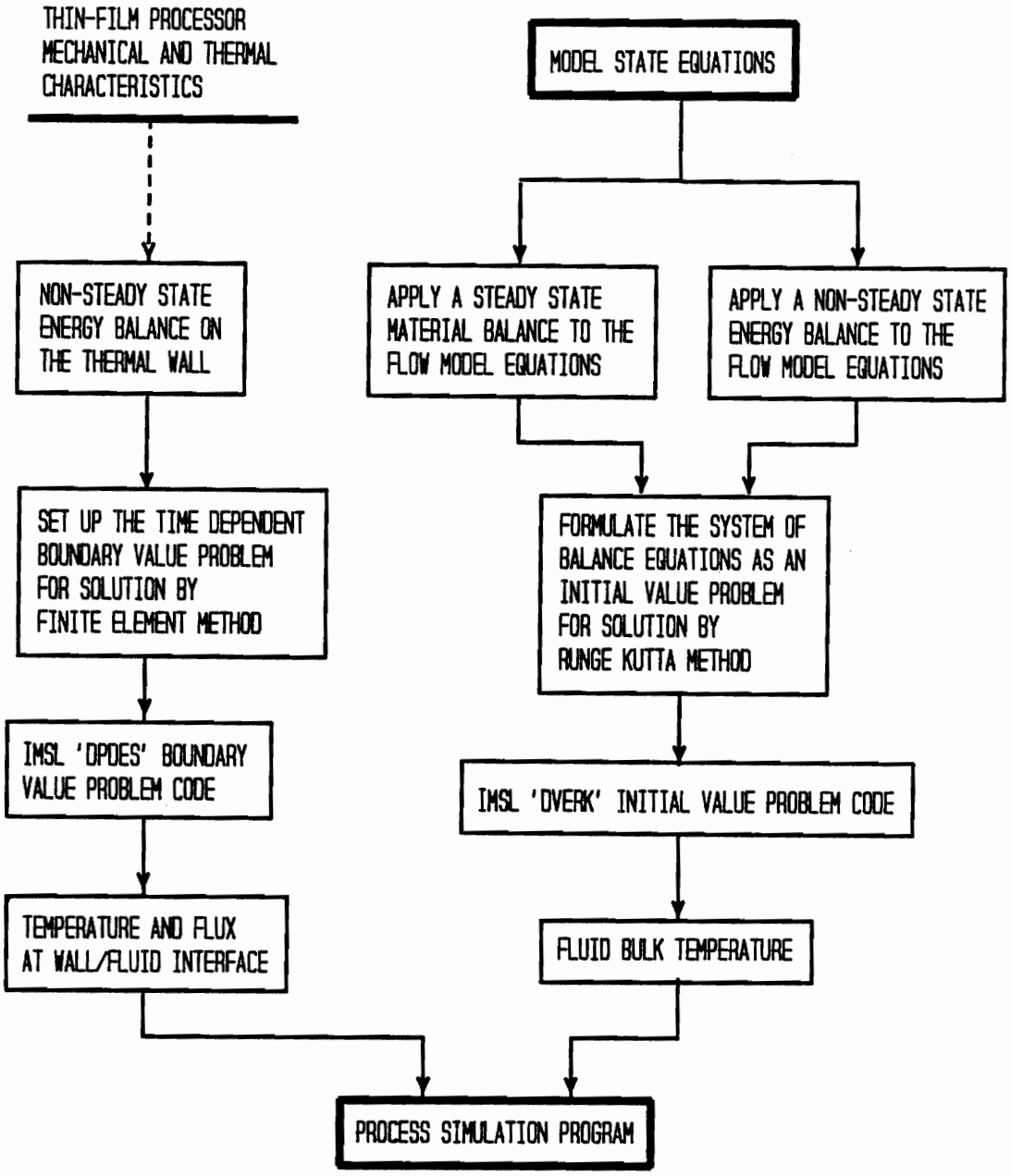


Figure 6.3. The development of a numerical code to simulate the heat transfer performance of the experimental processor. The block diagram is organized by tasks. The left branch includes tasks required to calculate the energy flux across the thermal wall; the right branch addresses the solution of the model state equations and integration of the energy flux into that solution.

at each stage.

Referring to Figure 6.3, the analysis of the thermal wall begins with the relevant mechanical and thermal characteristics of the thin-film processor, notably the thickness, thermal conductivity, density, and heat capacity of the wall. The wall surface area available to a single stage, at a specific set of operating conditions, is determined by the number of stages 'N' determined in chapter 3. Each of the 'N' non-steady state partial differential energy balance equations is assigned an independent call to the IMSL subprogram DPDES. Thus 'N' finite element, method of lines, codes integrate 'N' wall energy balances (one after the other stepping down the unit) over a prescribed time interval. The time-varying boundary conditions are supplied to each DPDES call through a user-supplied subprogram named BNDRY. DPDES returns the wall temperature distributions at the end of the user-specified integration interval.

Only *after* all the wall temperatures and fluxes are calculated for the interval are the 'N' fluid temperatures updated. The appropriate amount of fluid (as dictated by the overall process flow rate) is then transferred from each stage to its neighbor down the cascade. The task of updating the fluid temperature in each of the stages is managed by the IMSL initial-value problem code DVERK. DVERK employs a sixth-order Runge-Kutta code to solve the linear system of fluid temperatures as a function of time.

The processor heat transfer simulation program was written in Fortran and compiled on the IBM VS Fortran compiler (version 2, releases 1 and

2). The simulation was run on an IBM 3090 mainframe computer with typical execution CPU times of 3-10 minutes.

6.3.3 Coding Problems

A number of problems were encountered when writing the calling program for subroutines DVERK and DPDES, and their user-supplied subprograms. The following paragraphs describe the most significant problems and how they were resolved.

IMSL SUBROUTINE COMPATIBILITY: The subprograms DVERK and DPDES were not explicitly designed to be used together, to be called by the same program, or to pass variables between each other. To achieve smooth interaction between the codes it was necessary to re-initialize DPDES for each stage at the beginning of *every* time interval. This requirement greatly increased computation time as the number of model stages increased and the maximum allowable time interval (determined by the fluid residence time in a stage) simultaneously decreased.

THE MANAGEMENT OF TIME-VARYING BOUNDARY CONDITIONS: The process fluid and heating steam temperatures at the inside and outside of the heat transfer wall varied considerably over time, as can be seen from the plots in Appendix 3. The subprogram BNDRY, which supplies boundary conditions to DPDES, requires the functional time dependence of the steam and process fluid temperatures at their respective wall boundaries. DPDES does not apply the boundary conditions [6.17] directly. Instead, the

boundary conditions are imposed indirectly through the following differential equations (IMSL user's manual, 1984):

$$-h_O\frac{\partial T_w}{\partial t} + k\frac{\partial^2 T_w}{\partial t \partial x} = -h_O\frac{dT_O}{dt} \qquad \text{for } x=0, \text{ heating side}$$

$$h_I\frac{\partial T_w}{\partial t} + k\frac{\partial^2 T_w}{\partial t \partial x} = h_I\frac{dT}{dt} \qquad \text{for } x=L, \text{ process side} \qquad [6.18]$$

The constants h_O, h_I, and k are written into subroutine BNDRY by the user. Derivatives dT_O/dt and dT/dt are the time derivatives of the heating temperature and process fluid temperature respectively. They must be supplied as *continuous* functions of time. This is not possible for the process fluid derivative because the process temperature is calculated only after the inside wall temperature has been evaluated for each stage.

This contradiction was bypassed by creating a 'synthetic' process fluid derivative:

$$\frac{dT}{dt} = \frac{T-T_{last}}{\Delta t} \qquad [6.19]$$

The time derivative of the outside heating temperature was more difficult to generate. The application of a synthetic derivative caused the solution to become unstable and fail because of the steep, and rapidly changing, slope of the heating temperature curve near time zero. The problem was remedied by decomposing the heating temperature curves into a trigonometric series. The heating temperature curve was converted into an even function by joining it to a complementary set of points generated by a mirror reflection (see Boyce and DiPrima, 1977, Chapter 10). The resulting even function was duplicated several times to generate an even *periodic* function, as shown in Figure 6.4. This periodic function was then discrete Fourier transformed to yield the line spectrum in Figure 6.5 (Ramirez, 1985). The amplitude of each line represents a single Fourier coefficient in the Fourier series expansion of the heating temperature function. Some comments on Figure 6.5 are in order. First, the cosine coefficient at zero frequency has been omitted form the figure. Second, only the positive half of the full frequency spectrum is shown in the plots. Because the function is even, the sine coefficients at negative frequencies

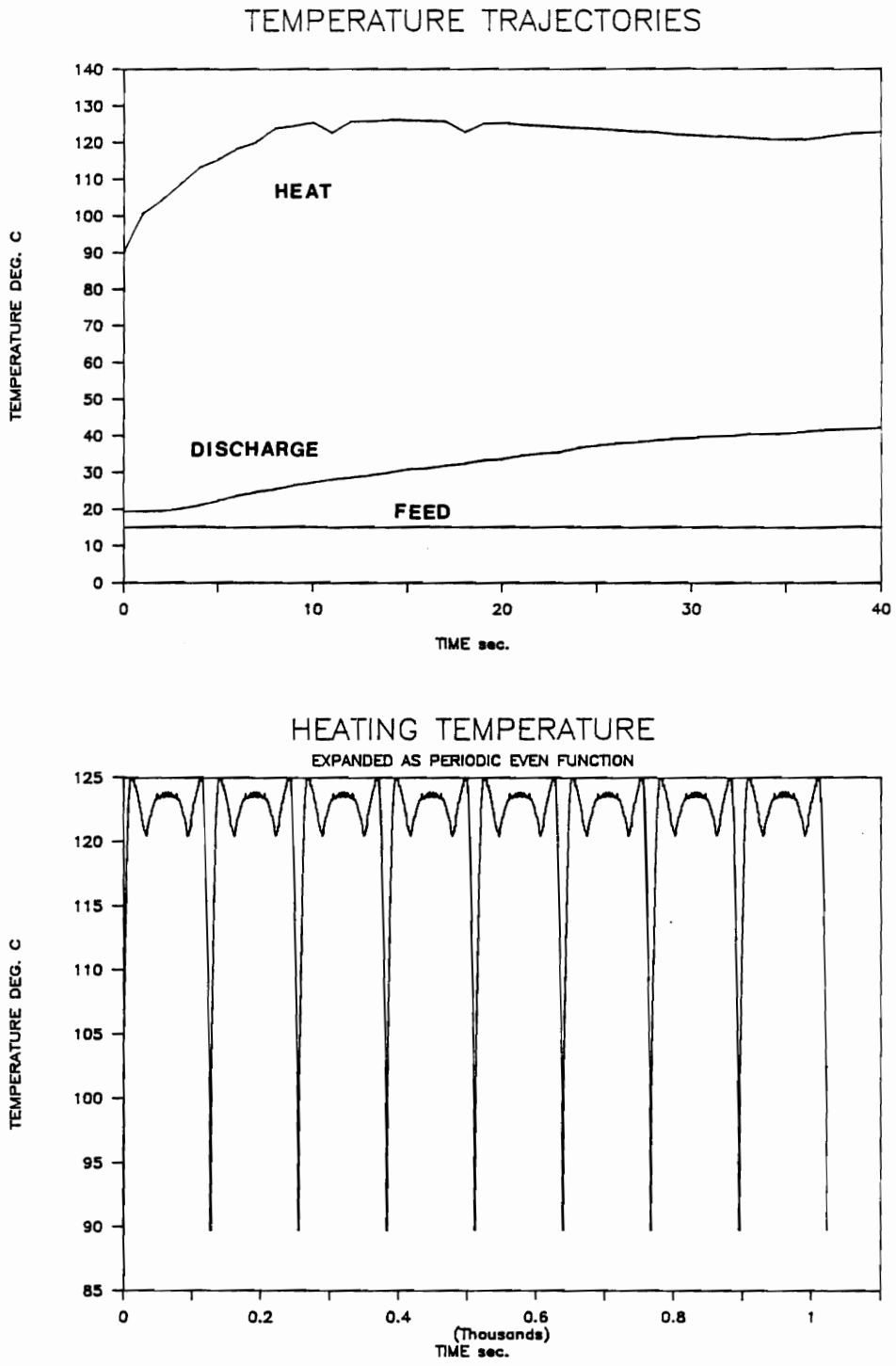


Figure 6.4. Preparation of the process heating temperature data for Fourier decomposition. - Heating, outlet, and feed, temperatures of an experimental run [top]. The experimental heating temperature repeated as an even periodic function for decomposition [bottom].

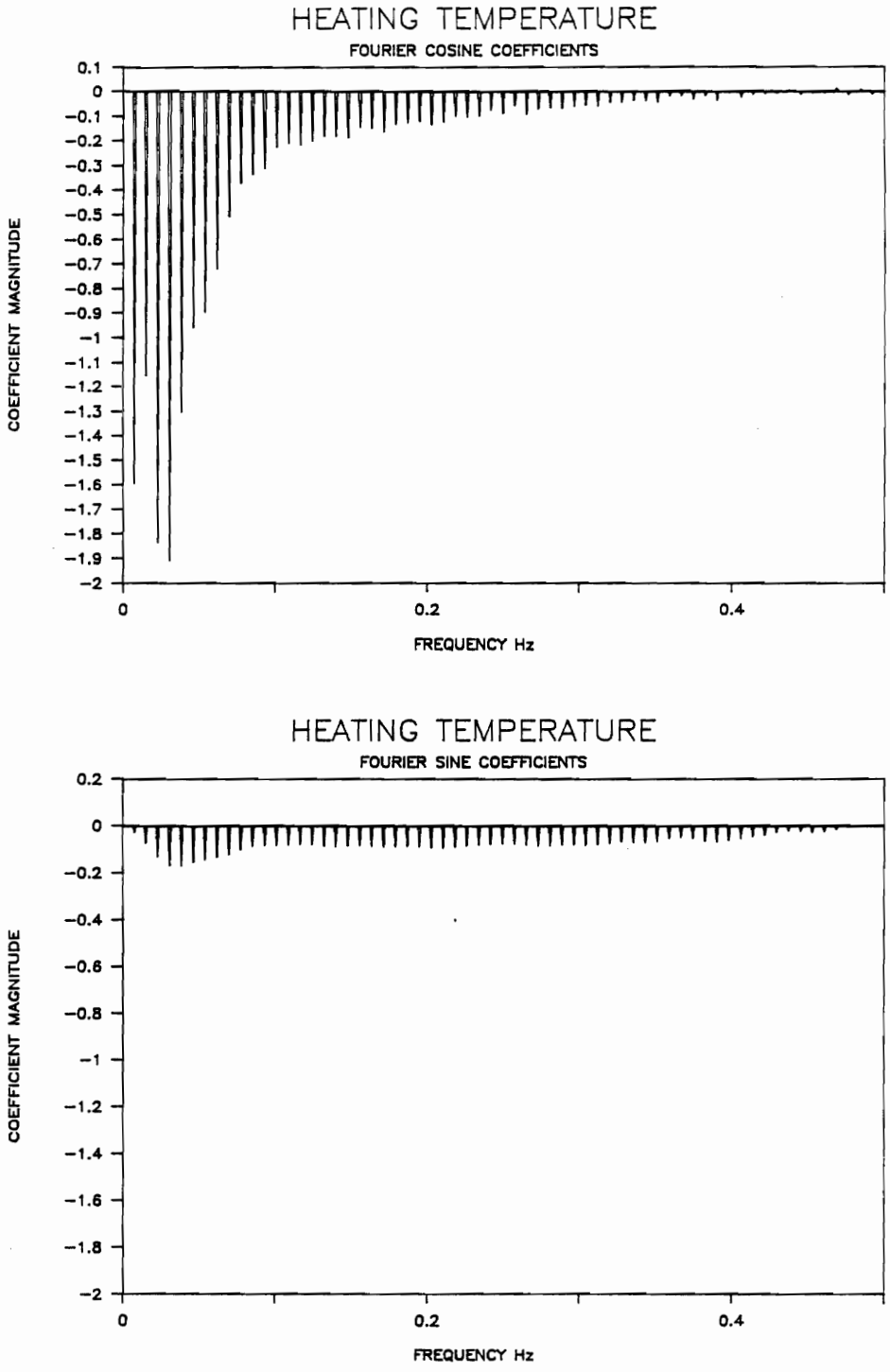


Figure 6.5. Fourier sine and cosine coefficients resulting from the transform of the even periodic heating temperature function in fig. 6.4. Note the zero frequency coefficient is omitted.

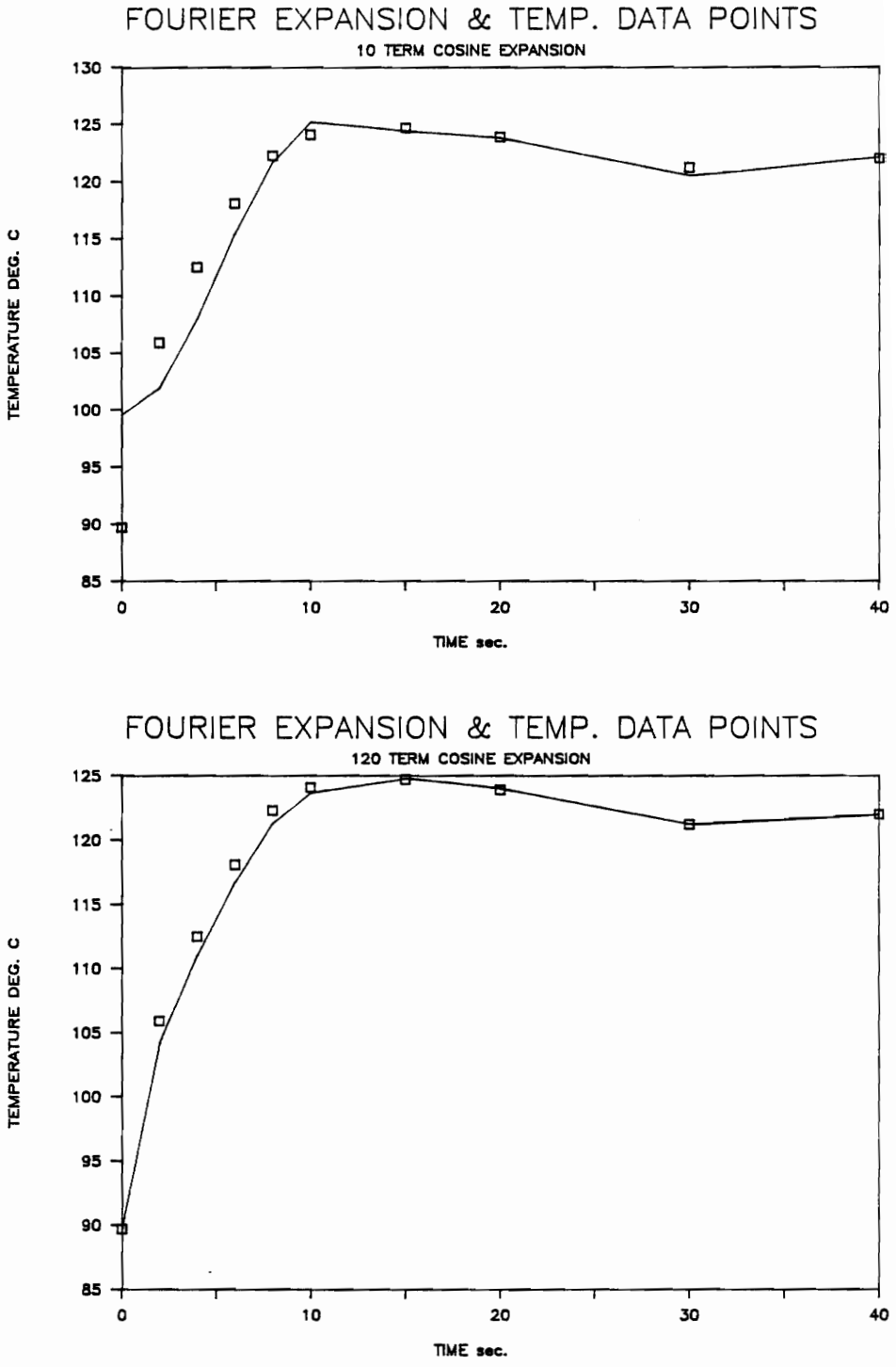


Figure 6.6. A comparison of the ten term and 120 term cosine expansions of the heating temperature data in fig 6.4. The squares represent experimental values.

are negative and cancel out their positive frequency counterparts. This leaves a simple cosine series.

The final cosine expansion of the experimental heating temperature curve was differentiated term by term to yield an analytic expression for dT_o/dt. The derivative was evaluated in a special subprogram TOPRIM. The number of terms actually evaluated was determined by the value of 'time'. At short times forty terms were used. The number of terms was decreased as the time increased and the slope of the curve became flat. The effect of the number of terms on the quality of curve replication is shown in Figure 6.6 where the squares represent experimental values. (Note: there are other, equally effective, approaches to generating a time function of the heating temperature. For example, interpolation from a look-up table.)

DETERMINATION OF ROTOR POWER: The net mechanical energy input to the process fluid by the rotor was calculated in the simulation for each set of operating conditions. This was considered preferable to experimentally measuring the electrical power consumption of the rotor drive motor because the motor was substantially over-sized. The calculation was made by assuming the difference between the inlet and outlet fluid temperatures, recorded prior to the introduction of heat at time zero, resulted exclusively from mechanical energy input by the rotor.

6.4 Running The Simulation

6.4.1 A Procedural Outline

Recalling that the simulation was developed to determine intrinsic process film coefficients, Figure 6.7 shows the information flow required for the transformation of transient experimental heat transfer data into an intrinsic or 'true' heat transfer film coefficient.

In Figure 6.7 the process simulation program generates an output temperature trajectory based on the experimental heating input function and a set of assumed inside and outside characteristic film coefficients. The validity of the assumed coefficients is checked by calculating, point by point, the sum-of-squares of the difference between the experimental and simulated temperature response of the processor. The procedure is completed when an acceptably good fit is obtained between the simulated and experimental transient temperature curves. Although the sum-of-squares error is a useful tool in making this determination, inspection of the curves, albeit subjective, was the final arbiter of acceptability.

The sum-of-squares errors were useful for generating successive guesses of the film coefficients as discussed below.

6.4.2 Searching For The Best Simulation-To-Data Fit

The set of sum-of-square errors (SSE's) for all possible choices of h_O and h_I can be viewed as points on a two-dimensional surface. The independent variables in this space are the inside and outside heat transfer film coefficients, h_I and h_O. Myers (1976) refers to this as a 'response surface' (not to be confused with *temperature* response). If a simple, locally valid, model can be proposed for this surface, then the location of minimum response can be predicted, that is, the h_O and h_I which give the closest fit between simulation and experiment.

The initial runs of the simulation revealed the actual response surface to be a very steep, curved, valley shown schematically in Figure 6.8. The response was locally modeled as a quadratic in both independent variables h_O and h_I:

$$SSE = \beta_o + \beta_1 h_O + \beta_2 h_I + \beta_{12} h_O h_I + \beta_{11} h_O h_O + \beta_{22} h_I h_I \quad [6.20]$$

The following procedure was used to predict values for the film coefficients that would yield the minimum SSE error:

1. The simulation program was run at nine selected points on the heat transfer plane of Figure 6.9, and nine SSE values were generated.
2. The six model parameters (β) were calculated from these nine SSE values, with three degrees of freedom, using standard linear

regression techniques.

3. The location of the minimum response was then estimated as follows:

$$\frac{\partial(SSE)}{\partial h_O} = \beta_1 + \beta_{12}h_I + 2\beta_{11}h_O = 0$$

$$\frac{\partial(SSE)}{\partial h_I} = \beta_2 + \beta_{12}h_O + 2\beta_{22}h_I = 0 \qquad [6.21]$$

In matrix notation:

$$\begin{bmatrix} 2\beta_{11} & \beta_{12} \\ \beta_{12} & 2\beta_{22} \end{bmatrix} \times \begin{bmatrix} h_O \\ h_I \end{bmatrix} = \begin{bmatrix} -\beta_1 \\ -\beta_2 \end{bmatrix} \qquad [6.22]$$

Restating,

$$\begin{bmatrix} h_O \\ h_I \end{bmatrix} = \begin{bmatrix} 2\beta_{11} & \beta_{12} \\ \beta_{12} & 2\beta_{22} \end{bmatrix}^{-1} \times \begin{bmatrix} -\beta_1 \\ -\beta_2 \end{bmatrix} \qquad [6.23]$$

The subsequent h_O and h_I calculated by [6.23] became the 'best guess' inputs for the next run of the simulation.

4. The steps above were repeated until an acceptable fit of the

simulation to experimental data was obtained.

Typical simulation fits are shown in Figure 6.9. A complete compilation of simulation-to-data plots is presented in Appendix 4.

These operations were carried out using the Statistical Analysis Services Institute (Cary, North Carolina) program library (SAS, version 5). The calling programs are listed in Appendix 6.

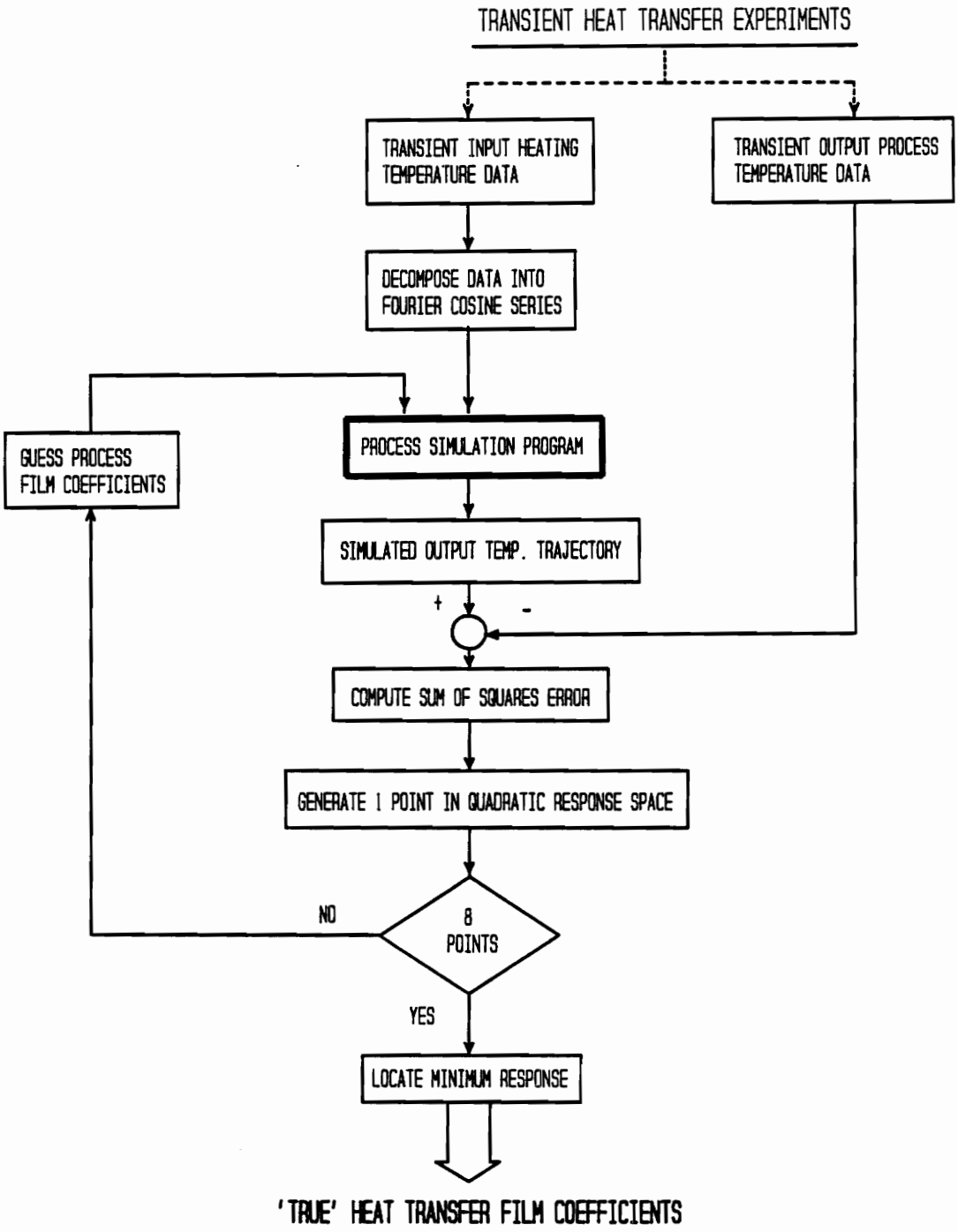


Figure 6.7. The protocol for converting experimental heat transfer data into process film coefficients. - The process simulation results were repeatedly compared to the corresponding experimental data and the program (inside & outside) film coefficients were adjusted in a statistical response space to achieve the best fit.

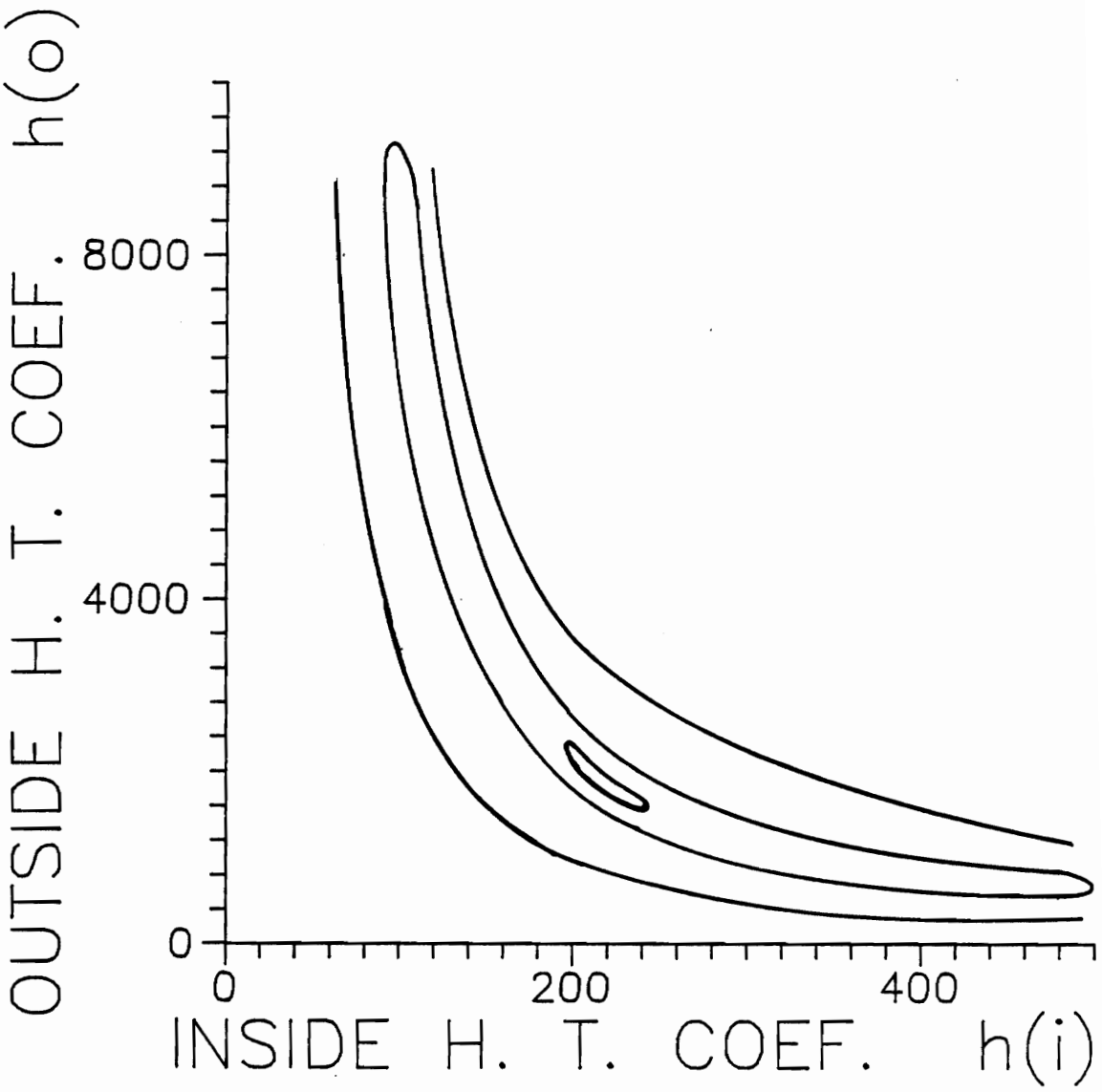


Figure 6.8. A schematic map of the inside and outside heat transfer film coefficient response surface generated by the protocol in fig. 6.7. The curves represent values of constant error between the simulated and observed transient temperature profiles (see fig. 6.9). The 'correct' film coefficient values are predicted at the minimum response. Note in this example the minimum response is located at h_i=220 and h_o=2000.

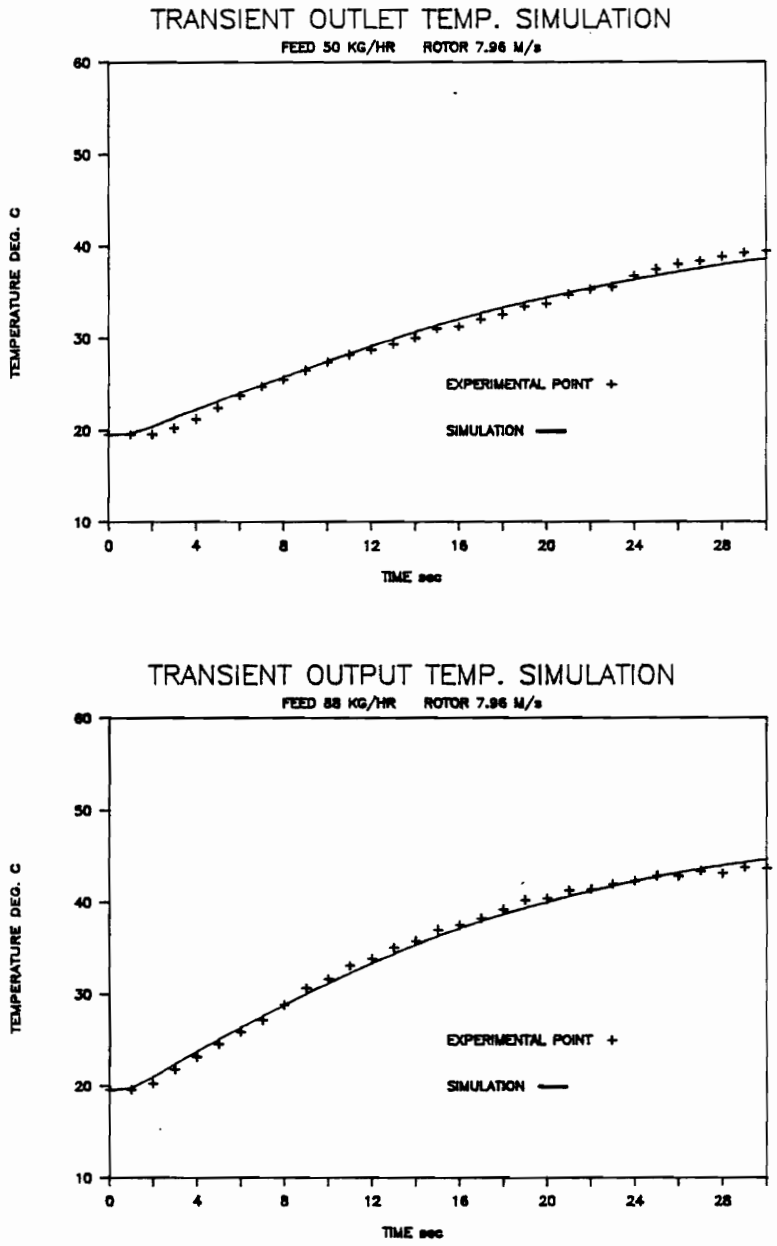


Figure 6.9. Typical experimental and simulated temperature trajectories are superimposed following the successful completion of the sum of squares error minimization procedure in fig. 6.7.

CHAPTER SEVEN

Heat Transfer Experimental Equipment And Procedures

7.1 The Heating Steam Control System

Heating steam was available at a pressure of 200 psig (1.46×10^6 Nm^{-2}) from the pilot plant main steam header. Steam was conducted from the header to the processor through a 1/2 inch nominal (~15.8 mm I.D.) pipe manifold. This manifold contained the heating steam pressure control system pictured in Figure 7.1. The system is a conventional pneumatic pressure control loop. The indicating pressure controller is a Foxboro Model 43A proportional integral controller with an integral pressure element and transducer. The pressure control valve is a 1/2 inch nominal (~15.8 mm I.D.) Badger Research control valve equipped with an equal percentage trim.

Steam condensate is normally voided from the processor heating jacket by a disk-type steam trap located below the heating jacket outlet nozzle.

The system is equipped with a manually operated ball valve (HV1) upstream of the processor steam inlet nozzle. This valve acts as an on/off blocking valve and can be used to isolate the processor from the steam manifold. A second manually operated valve (HV2) is installed between the processor steam outlet nozzle and the condensate trap. This valve allows the heating jacket to be vented directly to the atmosphere during

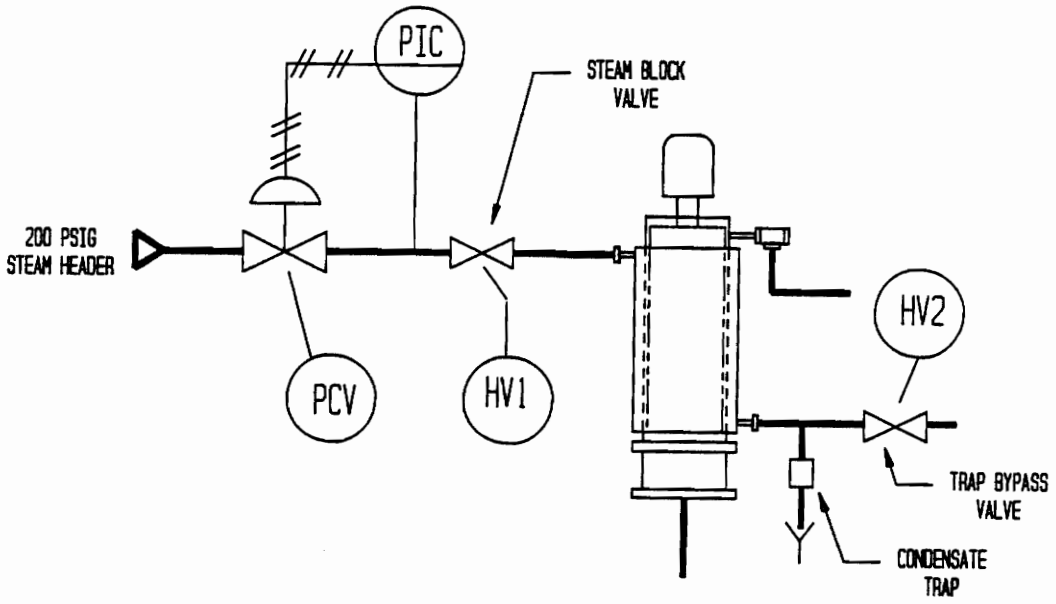


Figure 7.1. The heating steam pressure control system consisting of an indicating pressure transducer/controller and a pressure control valve. Hand valves HV1 and HV2 permitted rapid charging and clearing of the processor steam heating jacket.

start up, shutdown, and special operating situations.

7.2 The Process Temperature Measuring System

Chromel-alumel (Type K) thermocouples were the temperature elements used for all process temperature measurements. The thermocouples were configured with exposed, beaded junctions to minimize their response time. The thermocouples were purchased from Omega Engineering, Inc. under the following specification:

Thermocouple Type: K
Sheath: 1/4 in. 304 stainless steel
Junction: exposed, beaded type
Catalog number: CASS-14E-12

The thermocouple voltages were individually ice-point compensated, amplified, and scaled using one each Analog Devices Inc. AD595AD Type K thermocouple temperature transducer. The AD595AD has a delivered output of 10 mV/deg C. and an accuracy of ± 3 deg C. The transducer chip can be externally calibrated with the installation of a user-supplied resistor network. After this recalibration was performed, the output of all matched thermocouple/transducer pairs was within ± 0.3 deg C.

7.3 A Typical Heating Step Response Experiment

1. The condensate trap bypass valve HV2 was permanently opened to allow steam passing through the heating jacket to vent directly to the atmosphere. This was done for two reasons. First, to allow the rapid removal of non-condensable gases from the jacket during the early critical phase of the heating step change. Second, to insure that the heat transfer regime inside the heating jacket, i.e. the film coefficient, remained constant throughout the experiment. In this case, the heat transfer regime was primarily de-superheating.
2. The rotor speed was set at the desired value for the test. The heating steam block valve (HV1) was closed and the feed rate was set at a (high) value of 220 Kg/hr.
3. The processor was run for 15 minutes at these non-heating conditions to bring the processor heat transfer wall into thermal equilibrium with the feed liquid.
4. The feed rate was set to the desired value for the test.
5. The data logging computer was initialized and the run parameters were recorded. The data logging program was then started.
6. The heating block valve (HV1) was quickly opened (1/4 turn = full open), and the discharge temperature element was

placed in one of the process fluid rivulets falling from the circumference of the spool piece, and held there by hand. If the rivulet moved, the temperature element was moved to stay beneath it.

Note: The discharge thermocouple sheath was fitted with a tygon tip reservoir identical to that used with the conductivity probe (Figure 4.7).

7.4 Sources Of Experimental Error

7.4.1 Evaporation of the Process Fluid

During the transient heat transfer experiments some of the feed water inevitably evaporated. The evaporation rate was greatest at long times when the processor internal temperatures reached their maximum steady state values. These 'worst case' evaporation rates were measured experimentally. At steady state, evaporation accounted for about twelve percent of the heat transferred to the process fluid. Since the model energy balance does not take evaporation into account, heat transfer coefficients predicted by the simulation were low. However, the way in which the experiments were carried out minimizes the effect of evaporation on the simulation results by using only the first thirty seconds of the transient step response. The net amount of process fluid vaporized in this period is very small.

7.4.2 The Assumption Of Constant Film Coefficients

Average values of both process (inside), and heating (outside), heat transfer film coefficients were employed by the simulation. The actual coefficients are functions of temperature, the degree of superheat, the condensate film thickness, etc.. Variations in the process film coefficients were minimized by limiting the temperature change in the process fluid to 25°C. To insure that the heating steam film coefficient remained constant during the step input of heating steam, the steam jacket was vented directly to the atmosphere (by opening HV2). The high flow rate of steam that resulted kept the outside heating wall clear of condensate. An undesirable side effect of this procedure was the emergence of de-superheating as a major mode of convective heat transfer with resulting low film coefficients.

7.5 The De-superheating Problem

Film heat transfer coefficients for condensing steam typically reach values near 10^4 W m^{-2} K^{-1}. The values for de-superheating steam vapor, though, are one to two orders of magnitude lower (Holman, 1972). By opting for the de-superheating regime the following trade off was affected: a very low value was accepted for the heating film coefficient in exchange for a guarantee that the value would remain constant throughout the experiment. Because these low values for h_O are on the same order as the process side coefficients h_I, the thermal resistance of the heating side film cannot be neglected. A way must be found to evaluate both film coefficients

simultaneously from the experimental data. This was the motivation for the statistical analysis presented in Chapter 6.4.

CHAPTER EIGHT

The Handling Of Experimental Data

8.1 The Data Acquisition System

The equipment used for logging experimental data is shown schematically in Figure 8.1. The system employed commercial hardware and software components wherever possible. The application of custom designed components was restricted to the transducer level.

The system was built around the Laboratory Technology Corporation (Cambridge Massachusetts) LABTECH NOTEBOOK (R) data acquisition and management software package. The version (2.5) used in this work was written to support the MetraByte (R) Corporation model Dash 8 analog/digital converter running on the IBM Personal Computer under the DOS (2.1) operating system. These elements formed the core of the data acquisition system.

The responses of the thermocouple temperature elements (TE's) were detected, 'ice-point' compensated, and pre-amplified at the transducer level. These conditioned signals were sampled by the A/D converter on four separate channels selected through a programmable multiplexer. The sampling rate and total sampling time were user selected in software through the PC interface. The signal scaling and offset biasing functions were provided in software.

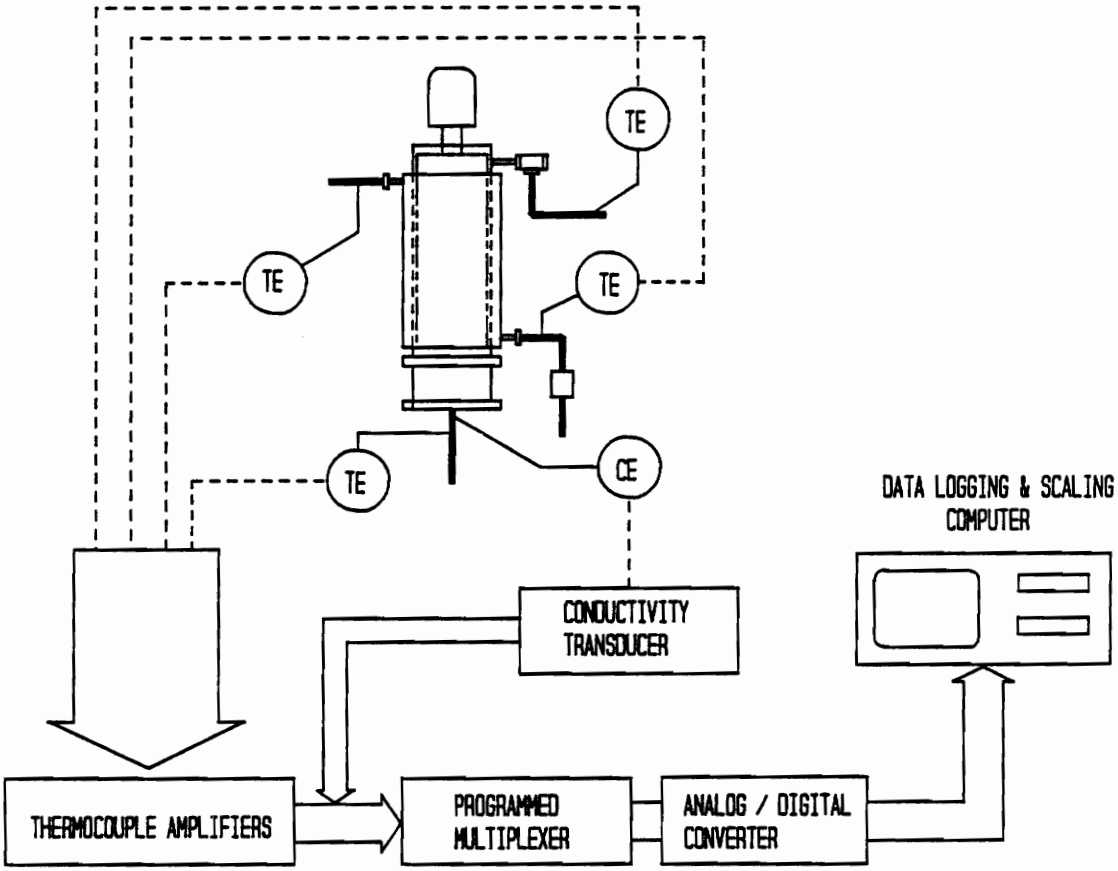


Figure 8.1. A block schematic diagram of the data acquisition and recording system. Thermocouples are represented in the figure as temperature elements (TE) and the conductivity probe as a conductivity element (CE). Note the temperature and conductivity experiments were run separately.

The response of the conductivity element (CE) was scaled at the transducer level, and did not undergo further conditioning in software.

While twelve bits of resolution are provided by the A/D converter, Labtech Notebook assigns a voltage of zero to the middle of the ten volt input range of the converter. This left eleven effective bits of resolution for signals that remain in the positive quadrant. The resulting resolution of one part in two thousand was more than adequate for this work.

8.2 Transducer Effects

8.2.1 The Convolution Of Transfer Functions

Each transducer has its own characteristic transfer function. Therefore, the response recorded by the data acquisition system is really the convolution of the processor response with a transducer response. Unless the response of the transducer can be shown to be negligible, it must be de-convolved from the recorded signal to obtain the 'true' processor response. Whether or not the transducer transfer function is negligible is, to some extent, a matter of judgment. A good empirical indication of the relative contributions of transducer and processor transfer functions is the magnitude of their characteristic time constants. The smaller the transducer time constant, the less its effect on the overall system response.

8.2.2 Conductivity Transducer Step Response

The time constant of an approximately first order transducer can be obtained from a step response experiment. The time constant is defined to be the time required for the transducer output to reach $(1-e^{-1})$ of its ultimate change in steady-state value, following a step change in input. The response of the conductivity transducer to a step change in tracer concentration is recorded in Figure 8.2. The experiment was carried out by wetting the conductivity probe tip with clean water then plunging it into a 2.5 percent potassium chloride solution. The time constant for the element-transducer system was approximately 0.1 second. This indicates significant attenuation for input signals with a frequency of 10 Hz or higher. How does this compare with the characteristic time constant of the processor? As the pilot plant was configured, a tracer step test was not possible. The processor time constant was estimated, instead, by substituting the fluid mean residence time in the processor. This time was consistent at about two seconds for all operating conditions. Thus, the transducer time constant can be seen to be one and one half orders of magnitude smaller than the overall system time constant. In this light it was considered safe to neglect the conductivity transducer transfer function in the analysis of experimental data.

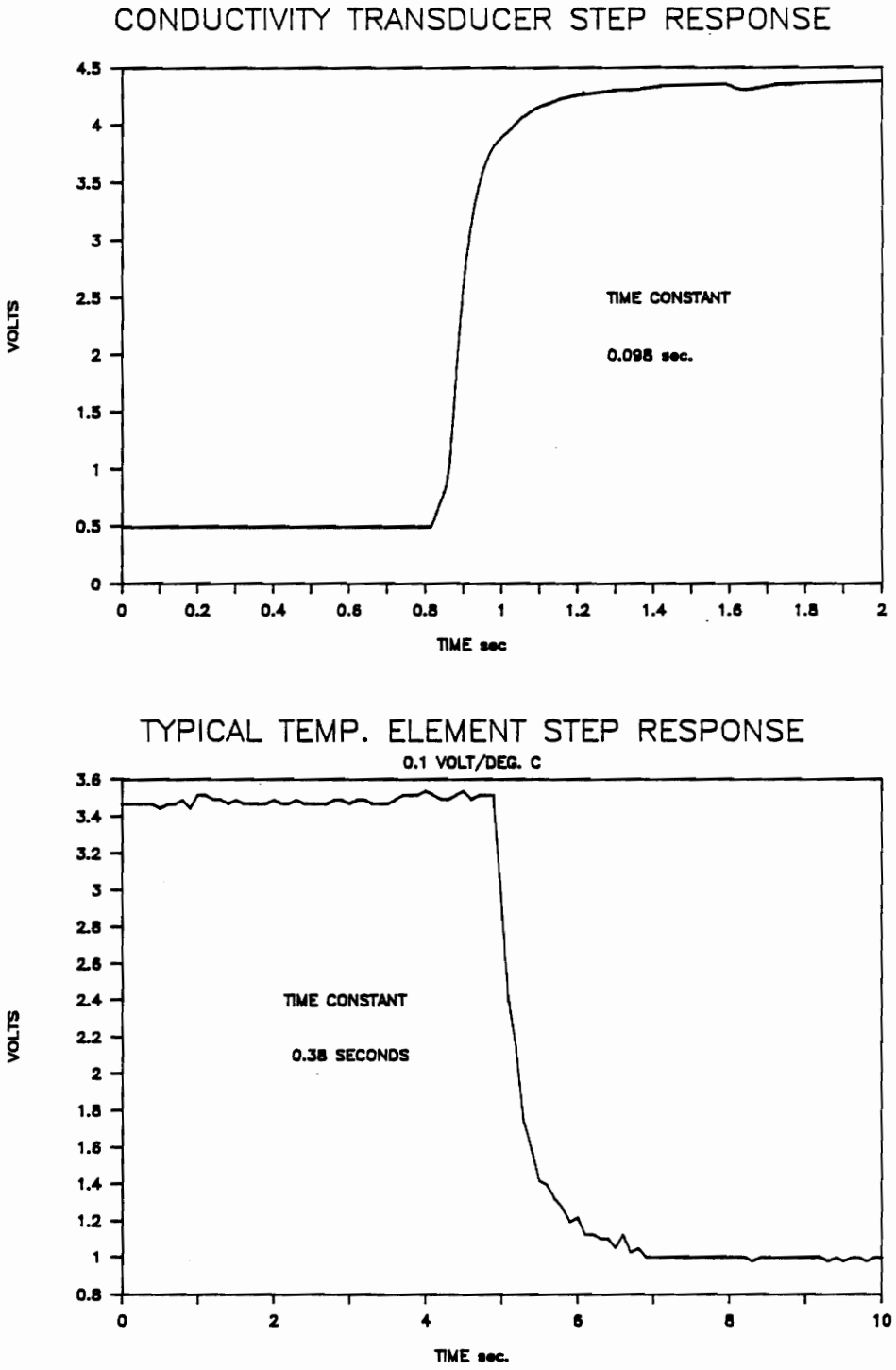


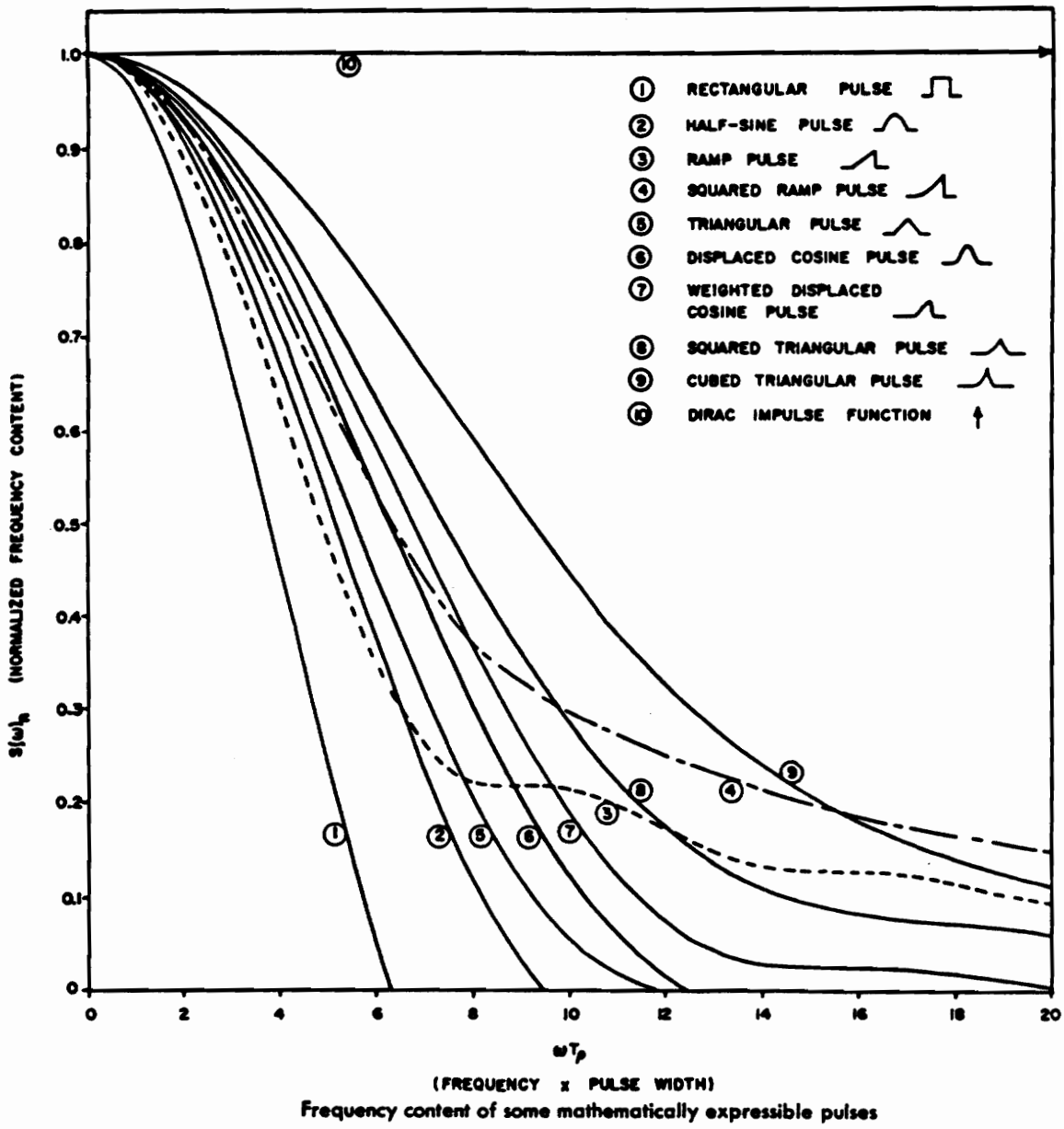
Figure 8.2. The dynamic step response of the conductivity transducer [top], and the temperature transducers [bottom]. - The time constants are .098 sec. and .38 sec. respectively.

8.2.3 Temperature Transducer Step Response

The time constant for a typical temperature transducer was determined similarly. Here the experiment involved plunging the thermocouple junction (initially in equilibrium with ambient air) into an ice bath and stirring. The results are shown in Figure 8.2. The transducer time constant was measured to be 0.38 seconds. Although the temperature time constant of the processor varied with operating conditions (see Appendix 3), it remained in the range of 20-50 seconds. Based on this difference of nearly two orders of magnitude, the transfer functions of the temperature transducers were neglected.

8.3 The Frequency Content Of The Input Pulses

Recalling that the parameters in the processor transfer function are to be evaluated in the frequency domain (section 3.3), some assurance must be obtained that the frequency content of the input pulse is uniform over the full response spectrum of the processor. An ideal impulse input meets this requirement for *any* system because it, by definition, contains all harmonic components from zero Hertz to infinity at equal amplitude. Real, or approximate, impulse inputs do not! The amplitude of the input signal decreases rapidly for frequencies above a certain cutoff. The cutoff



Frequency content of some mathematically expressible pulses

Figure 8.3. The frequency content of different idealized input pulses are pictured here. The weighted displaced cosine pulse (7) most closely represents the hand held injections of the tracer experiments.

frequency itself is defined as the frequency at which the amplitude has decreased to 90 percent of the zero frequency (or D.C.) level. The frequency content of the experimental tracer injections was estimated using the frequency content diagram in Figure 8.3. This figure plots amplitude versus frequency for a number of mathematically expressible pulses. The experimental injection pulse width was estimated to be 0.1 seconds. The pulse shape that best describes the injection by a hand held syringe is the weighted displaced cosine in which the injection rate increases steadily as the plunger is accelerated, then ends abruptly when the plunger reaches the end of the barrel. At the cutoff frequency, where Figure 8.3 shows a normalized frequency content of 0.9, the product of frequency × pulse width is 3. For a pulse width of 0.1 it indicates the highest usable frequency contained in the injection is 3/0.1 = 30 Hz. The observed *injection* response of the processor is therefore equivalent to its ideal unit impulse response for frequencies below 30 Hz.

8.4 The Sampling Rate And Aliasing

The frequency content of the input pulse places a logical upper limit on the sampling rate of the experimental output signal. Any output component measured above thirty Hertz will not reflect the processor input response but will, instead, reflect unrelated events such as system noise.

There also exists a lower limit on the sampling rate. The lowest acceptable sampling rate is determined by a phenomena known as aliasing. Aliasing occurs when a time sampled signal contains harmonic components with frequencies greater than:

$$\frac{1}{2\Delta t} \qquad \text{where } \Delta t = \text{the sampling interval} \qquad [8.1]$$

This is formally known as the Nyquist frequency. At frequencies above the Nyquist frequency, the frequency components cannot be unambiguously assigned. The result is that such components are reflected back into the spectrum at a frequencies below the Nyquist limit. This is shown graphically in figure 8.4 where signal power, the square of signal amplitude, is plotted versus frequency. Figure 8.4 shows the effect of distortion caused by aliasing of the frequency spectrum of a time sampled signal. To avoid such distortion the highest frequency component in the sampled signal must reside below the Nyquist frequency.

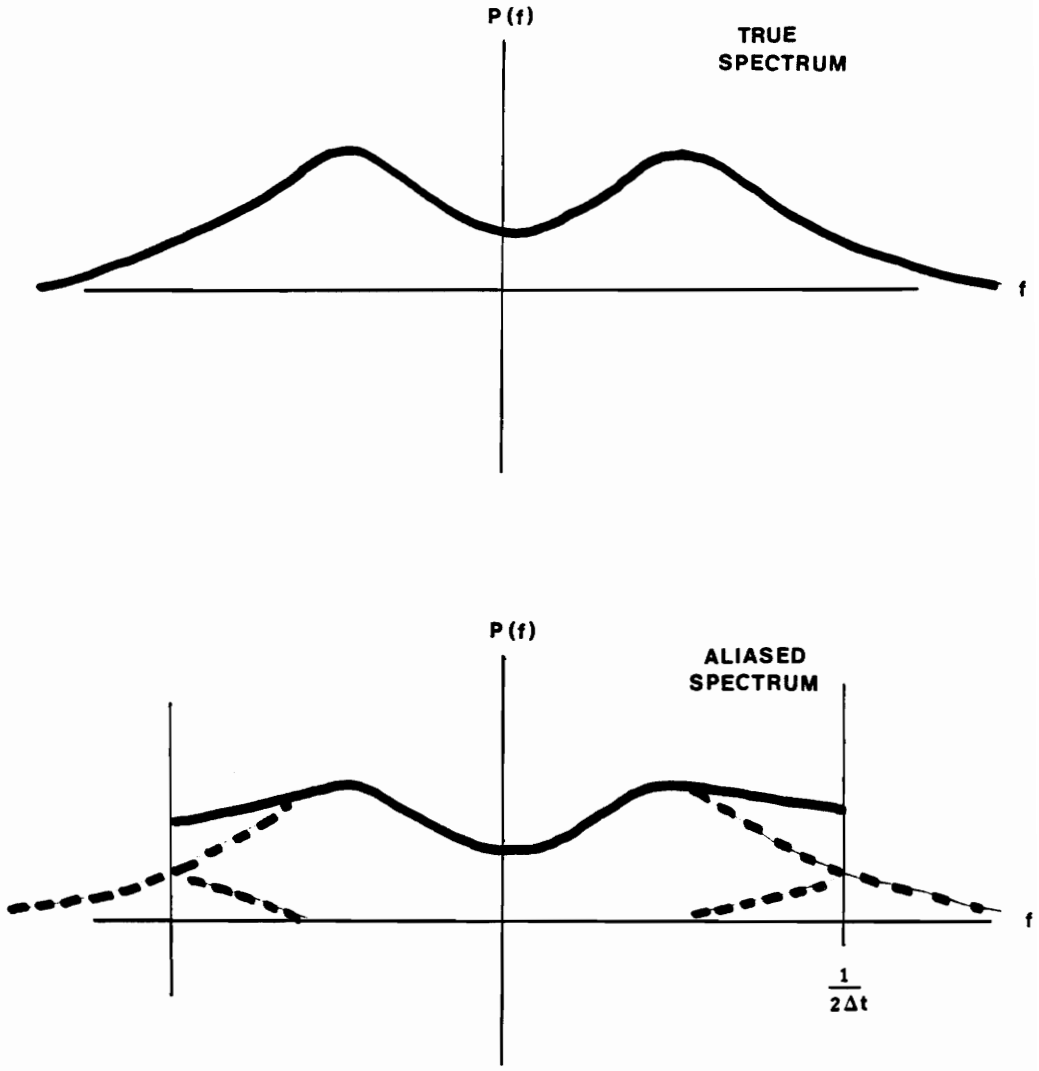


Figure 8.4. The true and aliased representations of a hypothetical signal power spectrum. Distortion of the aliased signal results from power reflected into the spectrum from high (un sampled) frequencies. The true spectrum can be obtained by increasing the sampling rate. Parameters estimated in the frequency domain using aliased data will yield an invalid model.

Since the highest frequency component of the thin-film processor's response was not known a priori, an arbitrary sampling interval of 0.05 seconds was initially used, giving a Nyquist frequency of 10 Hertz. The time domain response of an injection at these conditions is shown in Figure 8.5. The corresponding response in the frequency domain is shown in Figure 8.6. It is noteworthy that, by 5 Hertz, both the real and imaginary parts of the processor's frequency response go to zero. This offers a substantial safety margin with a Nyquist frequency of 10 Hertz.

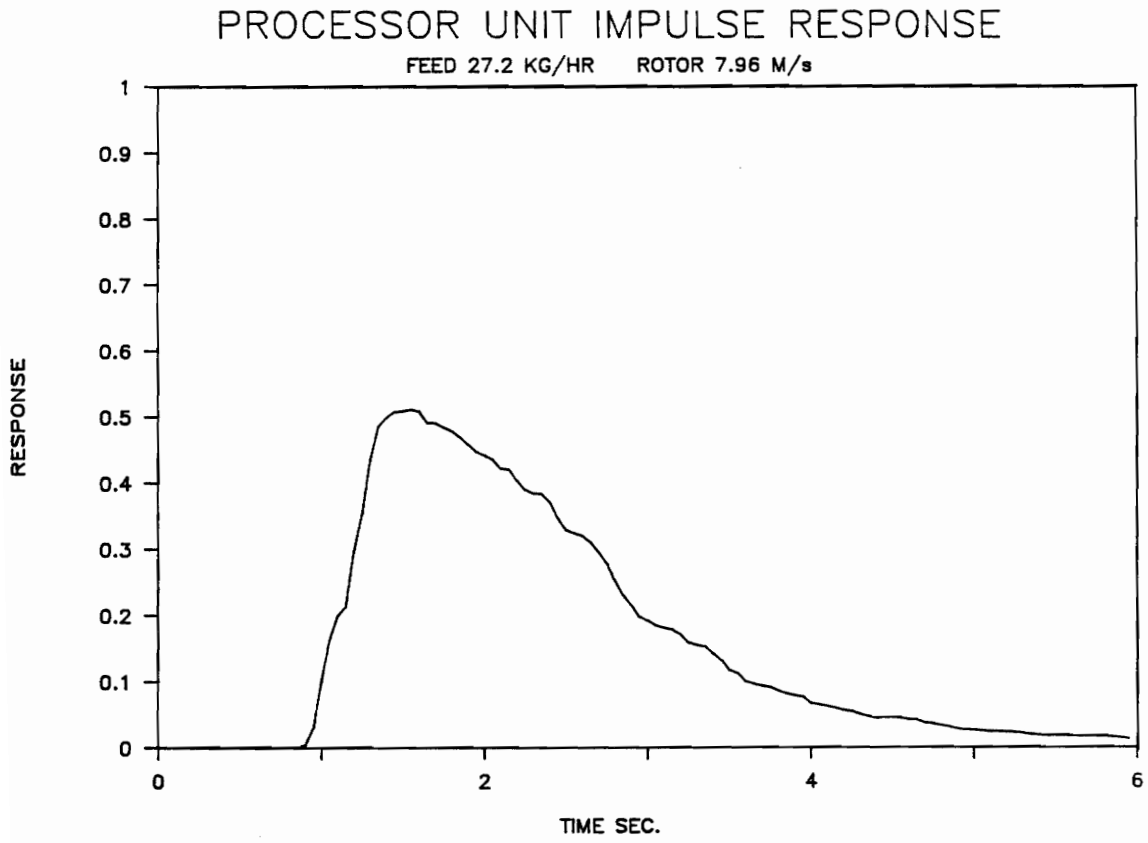


Figure 8.5. The time domain unit impulse response of the processor for a typical tracer injection experiment.

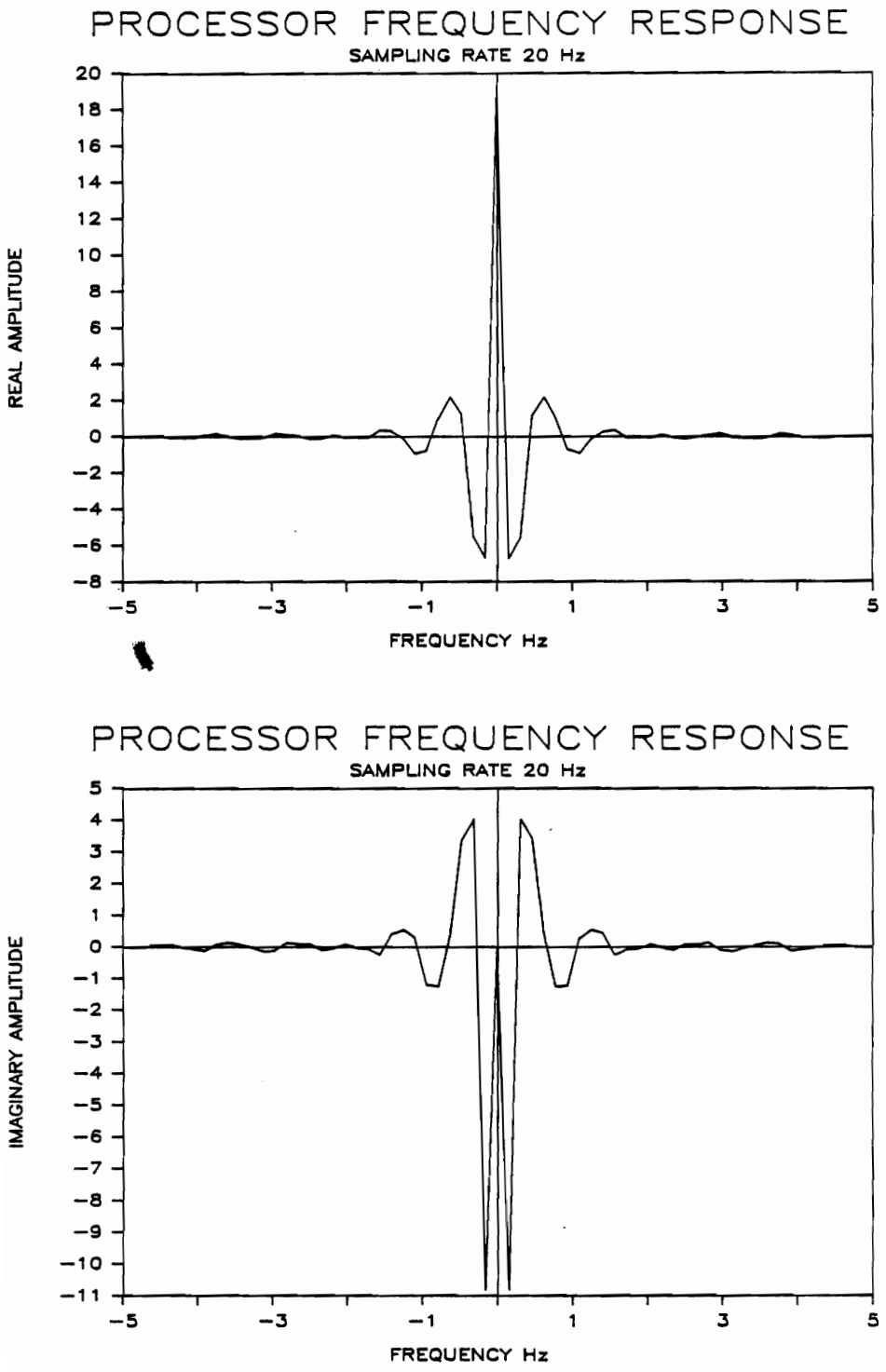


Figure 8.6. The Fourier transform of the data in fig. 8.5 into the frequency domain. The real [top] and imaginary [bottom] parts of the response are plotted separately. Note the absence of aliasing at the sampling rate of 20 Hz (Nyquist frequency 10 Hz).

8.5 Post Acquisition Signal Processing

8.5.1 Raw Data Conditioning

The output data from the tracer injection experiments required three conditioning steps before being applied to model formulation or analysis.

1. CORRECTION: The conductivity data was corrected for the non-linear relationship between tracer concentration and transducer voltage. The correction function was a fourth order polynomial in transducer voltage:

$$\text{Wt.\% KCl} = 0.0084\ V^4 + 0.07424\ V^3 + 0.2022\ V^2 + 0.1903\ V + 0.04952 \qquad [8.2]$$

2. NORMALIZATION: Each data set was normalized by first numerically integrating the area under its response curve. Each point in the data set was then divided by the area to transform it into a point on the normalized data set.

Normalization of the response was required for two reasons. First, normalization eliminates the effects of variations in the amount of tracer injected; it effectively makes all injections equal. Second, the normalized impulse response curve is, by definition, the system's unit impulse response. It is identically equal to the thin-film processor's residence time distribution function.

The fluid mean residence time τ is the point, on the time axis, which equally divides the area under the residence time distribution curve, one half to the left and one half to the right. Normalized response curves for all the experimental points are plotted in Appendix 1.

3. AVERAGING: The normalized responses were subsequently filtered to improve the signal-to-noise ratio, as described in the signal averaging section below.

The output data from the transient heating experiments required only filtering to prepare it for use with the temperature simulation.

8.5.2 Signal Averaging

Significant levels of noise were present in the data from both the tracer injection and the transient heating experiments. Different approaches were considered to deal with the noise. The first to be considered was a class of post processing procedures known as moving-average algorithms. These algorithms digitally convolve a data set with a selected convolution function. The simplest convolution function is a rectangular pulse of unit height and its associated time average is known as a 'box-car average'. The signal-to-noise ratio for moving average techniques is proportional to the square root of the number of data points spanned by the convolution function. These algorithms share the ability to attenuate both high and low frequency noise components. They also require only a single spectrum (data set), which is a definite advantage for difficult or time consuming

experiments (Dessy, 1978, p. 157-167).

An alternative approach is offered by ensemble averaging techniques. These techniques operate on a collection of discrete spectra taken for some repeated event. The ensemble average value of a data point 'k' at a given time 't' over 'M' spectra is the sum of all equivalent points 'k' divided by 'M', where 'M' is the number of times the experiment was repeated. The signal-to-noise ratio for ensemble averaging is proportional to the square root of the number of ensembles (response curves) used. Like the moving average methods, ensemble averaging filters both high and low frequency components of noise (Dessy, 1978, p. 161-165).

A word about noise itself is in order. The high frequency noise in the tracer injection spectra was primarily the result of statistical fluctuations associated with the fluid convection inside the processor. The low frequency noise was due to slight variations in the experimental injection technique. The importance of removing the latter is evident. The heating response experiments were plagued by severe high frequency fluctuations, but low frequency noise was not evident.

Simple ensemble averaging was used for all experimental work. This selection was based on several considerations. Ensemble averaging calculations are easily implemented in a spread sheet format. The accumulation of multiple responses provided additional information about the variability of the process, the variability of the experimental method, and the consistency of the experimenter.

Implementation of ensemble averaging for the tracer injection experiments involved repeating each test point three times. The three data sets were then added together point for point and divided by three. The theoretical improvement in signal-to-noise ratio of $(3)^{1/2} = 1.73$ does not appear very significant, but Figure 8.7 shows the improvement is in fact quite dramatic. The transient heating experiments required four ensembles to satisfactorily attenuate statistical fluctuations (Figure 8.9).

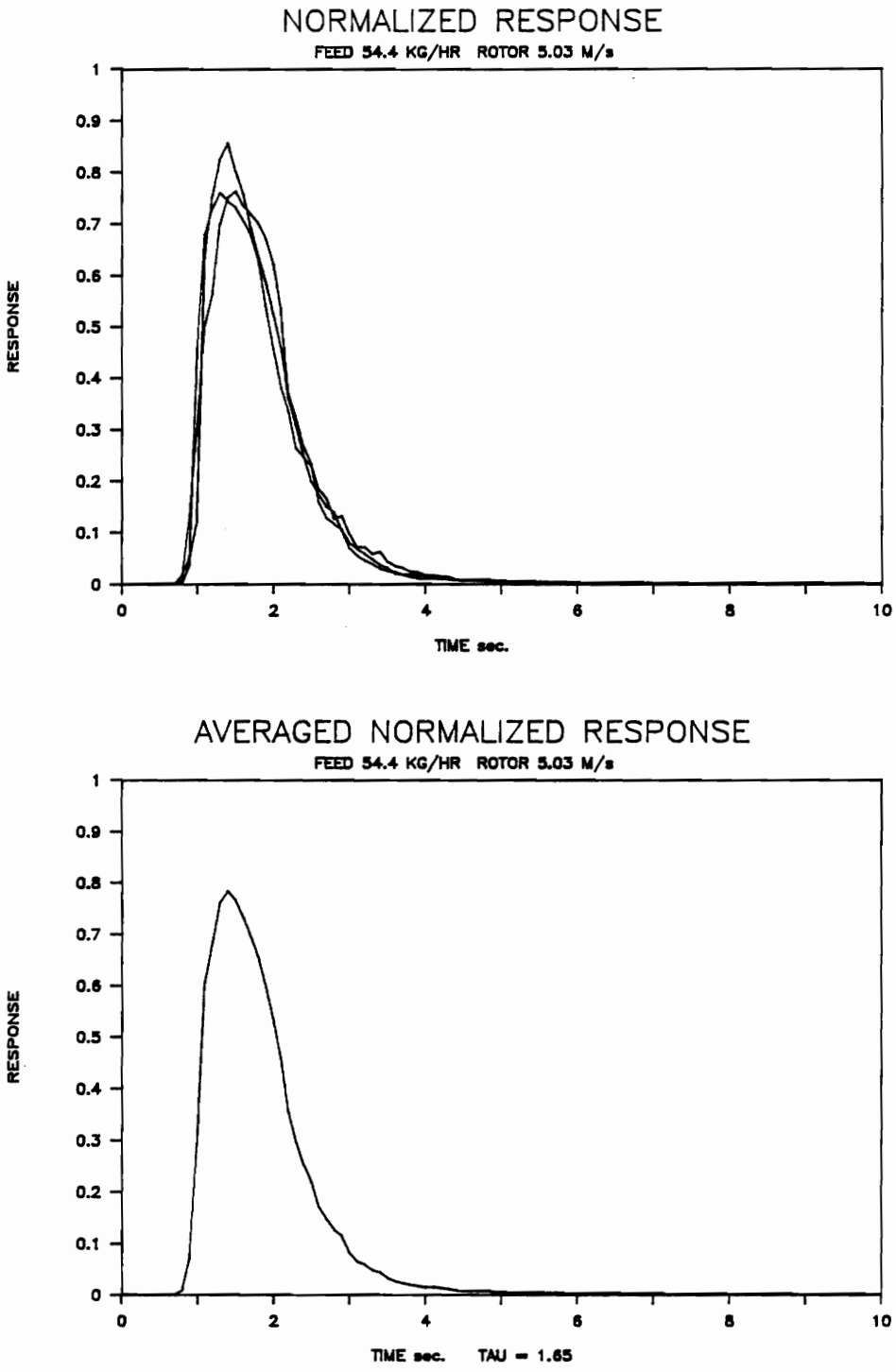
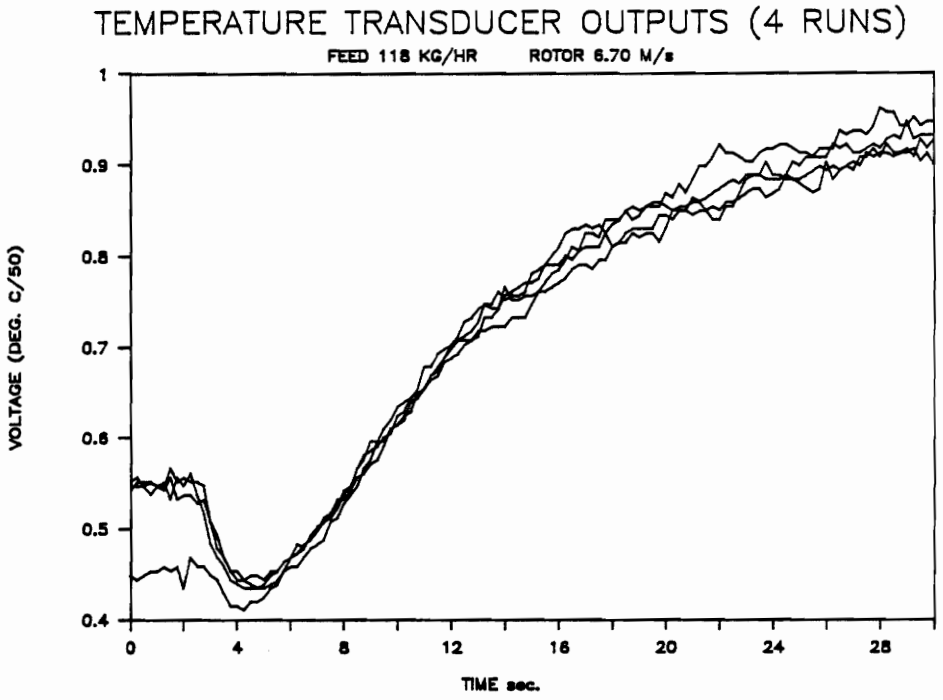


Figure 8.7. The result of ensemble averaging of a typical set of impulse response curves. TAU is the mean residence time.



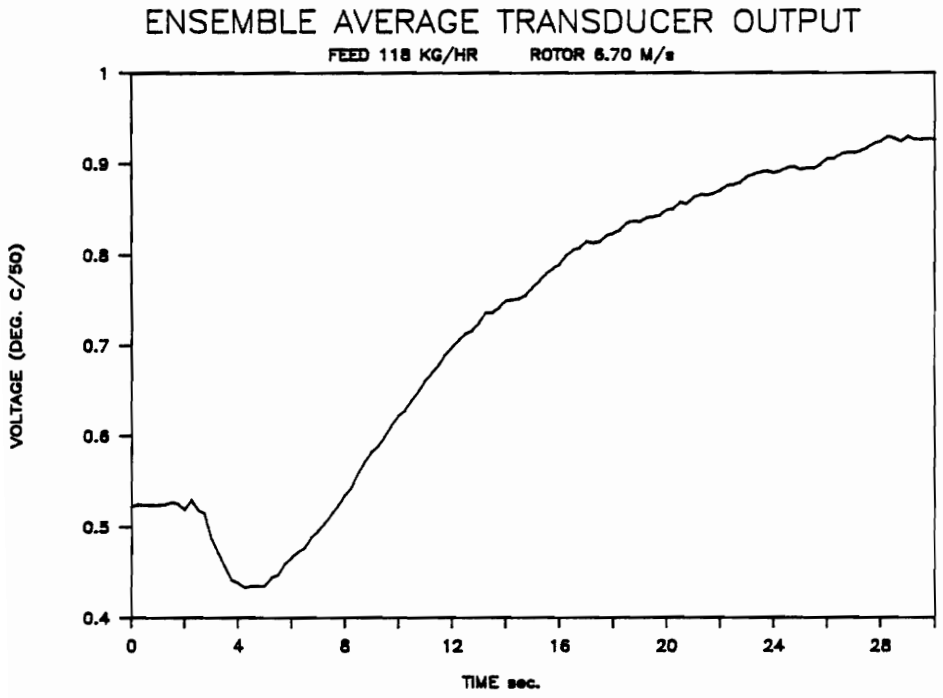


Figure 8.8. Ensemble averaging results for a typical heat transfer experiment.

CHAPTER NINE

Discussion of the Experimental Results

9.1 Quantitative Evaluation of the Models for Fluid Flow

9.1.1 Introductory Comments

The numerical results of the tracer impulse response experiments are provided in Table 9.1. This tabulation is organized by model for the tanks in series, and the series with dead time models. Before examining the results for each model, some comments are offered on the general aspects of model fitting. Both models were fitted to experimental data using the minimum sum-of-squares error (SSE) criteria given in section 6.4.2. While the SSE method generates those parameters that produce the best overall fit of the model to a corresponding data set, it does not allow quantitation of the physical correctness or engineering suitability of a model. These are subjective determinations that must be made by the investigator(s). In this work they involved a critical comparison of the experimental results with those obtained from a simulation of the model. Visual examination of the experimental and simulation curves, the SSE values, and engineering judgment were all used. The assessment was that the tanks in series model

TABLE 9.1

PARAMETER ESTIMATION RESULTS

FEED kg/hr	Θ_{EXP} sec.	TANKS IN SERIES MODEL N	τ sec.	SSE	MODEL WITH DEAD TIME N	Dt sec.	τ sec.	SSE
			ROTOR SPEED 5.03 m/s					
27.2	1.95	8	.250	3.1	3	.800	.414	.8
54.4	1.65	10	.170	3.6	3	.787	.328	.4
71.1	1.75	15	.119	2.7	4	.856	.247	.2
90.7	1.95	18	.110	9.9	4	1.027	.254	3.5
102	2.00	17	.117	1.0	11	.415	.145	1.1
118	2.05	24	.086		4	1.208	.232	1.5
132	2.10	31	.072	14.2	4	1.313	.208	
			ROTOR SPEED 6.70 m/s					
27.2	2.05	8	.250	14.6	2	.881	.659	.5
52.2	1.75	10	.175	14.9	2	.925	.490	3.4
72.6	1.80	14	.130	13.0	4	.850	.263	5.6
90.7	2.00	18	.112	6.8	4	.875	.260	1.1
104	2.15	22	.100	5.0	6	1.063	.197	.9
118	1.95	20	.100	4.0	7	.800	.176	2.2
132	2.10	24	.088	5.6	7	.938	.170	1.6
			ROTOR SPEED 7.96 m/s					
27.2	2.10	8	.266	12.0	2	.975	.680	2.1
54.4	1.65	8	.215	7.1	4	.550	.304	1.6
70.3	1.80	14	.131	5.5	4	.825	.269	1.8
88.4	1.95	16	.125	8.0	4	.981	.273	3.0
102	2.00	21	.097	7.4	5	1.000	.218	2.0
118	2.10	17	.125	9.0	4	1.081	.277	2.4
132	2.20	20	.113	3.7	6	1.000	.216	1.2
			ROTOR SPEED 8.80 m/s					
54.4	2.00	7	.300	5.3	3	.650	.506	2.2
70.3	2.00	13	.156	9.9	3	.994	.380	5.0
88.4	2.05	14	.152	4.9	5	.834	.269	3.2
102	2.20	19	.119	3.9	5	1.075	.246	1.1
118	2.30	22	.106	6.5	4	1.300	.280	0.3
132	2.30	21	.113	8.1	4	1.325	.276	2.1

Θ represents mean residence time; N the number of stages; τ residence time per stage; Dt dead time; SSE sum of squares error.

provides a 'reasonable' representation of the state of fluid convection inside the thin-film processor, and that the addition of dead time to the model provides a 'good' representation.

When using a model's ability to simulate experimental data as an 'objective' criteria of model suitability, it should be remembered that the model parameters are not independent constants. Rather, they are systematically selected to minimize the deviation between the model and a data set. Thus, many different models might give reasonably good experimental curve fits. The two models proposed for this work provided a consistent data fit over a wide range of operating conditions, lending some credibility to their underlying physical assumptions.

9.1.2 The Tanks in Series Model

The advantage of this model is its physical simplicity. The disadvantage is the model's poor fit to experimental data in two regions of the impulse response / residence time distribution curve, specifically, the initial processor response at about one second and the trailing tail of the response at 3-4 seconds. Figure 9.1 and Table 9.1 show the model fit is generally weak at low feed rates and improves steadily until about 100 kg/hr, deteriorating somewhat at higher rates. Inspection of the curves in Figure 9.1 indicate a reduction in error could be achieved by shifting the model response along the time axis. When a dead time is incorporated in the tanks in series model, a significant improvement in the fit results. This is seen in

both the SSE values in Table 9.1 and a striking visual improvement in the model fit in Figure 9.3.

The data compilation in Table 9.1 poses the question: what is the relationship between the number of stages and the feed rate? Plotting model stages versus feed rate in Figure 9.2 shows the number of stages to be an approximately linear function of feed rate. The slope decreases with rotor speed, except at the highest rotor speed of 8.8 m/s.

These linear curve fits were used to determine the appropriate number of stages to code into each of the heat transfer simulations.

9.1.3 The Tanks in Series Model With Dead Time

Figure 9.3 shows the model fit for the tanks in series model modified to include a dead time. Visual inspection of the curves indicates a much better representation of the process. This is born out by the sum-of-squares error terms in Table 9.1. A physical interpretation of the dead time is obscured by the fact that the model is linear. That is, the model is constructed from linear elements. These elements obey the associative and distributive laws of linear operations. Since the model is linear, the dead time can be placed anywhere within the cascade and still generate the same response. Therefore, this does not offer any insight into the physical interpretation of the dead time. This is illustrated in Figure 9.5. Figure 9.5a is a linear flow model with the dead time represented

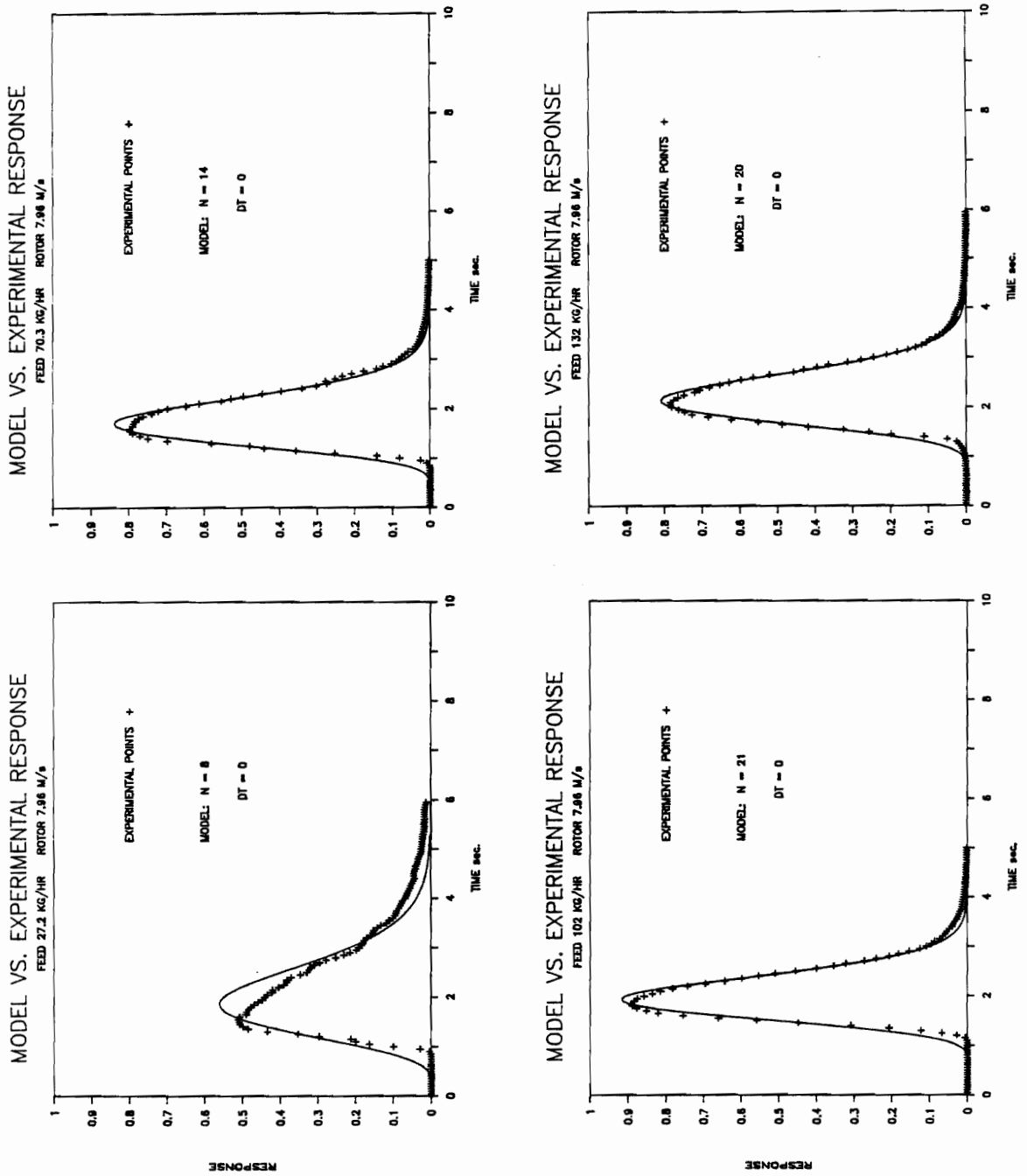


Figure 9.1. Fitting the straight tanks in series model to experimental data (at one rotor speed). The solid curves represent model predictions. Parameter 'N' is the number of ideal model stages. In these models the dead time 'Dt' is zero. - A complete catalog of model fits is presented in appendix 2.

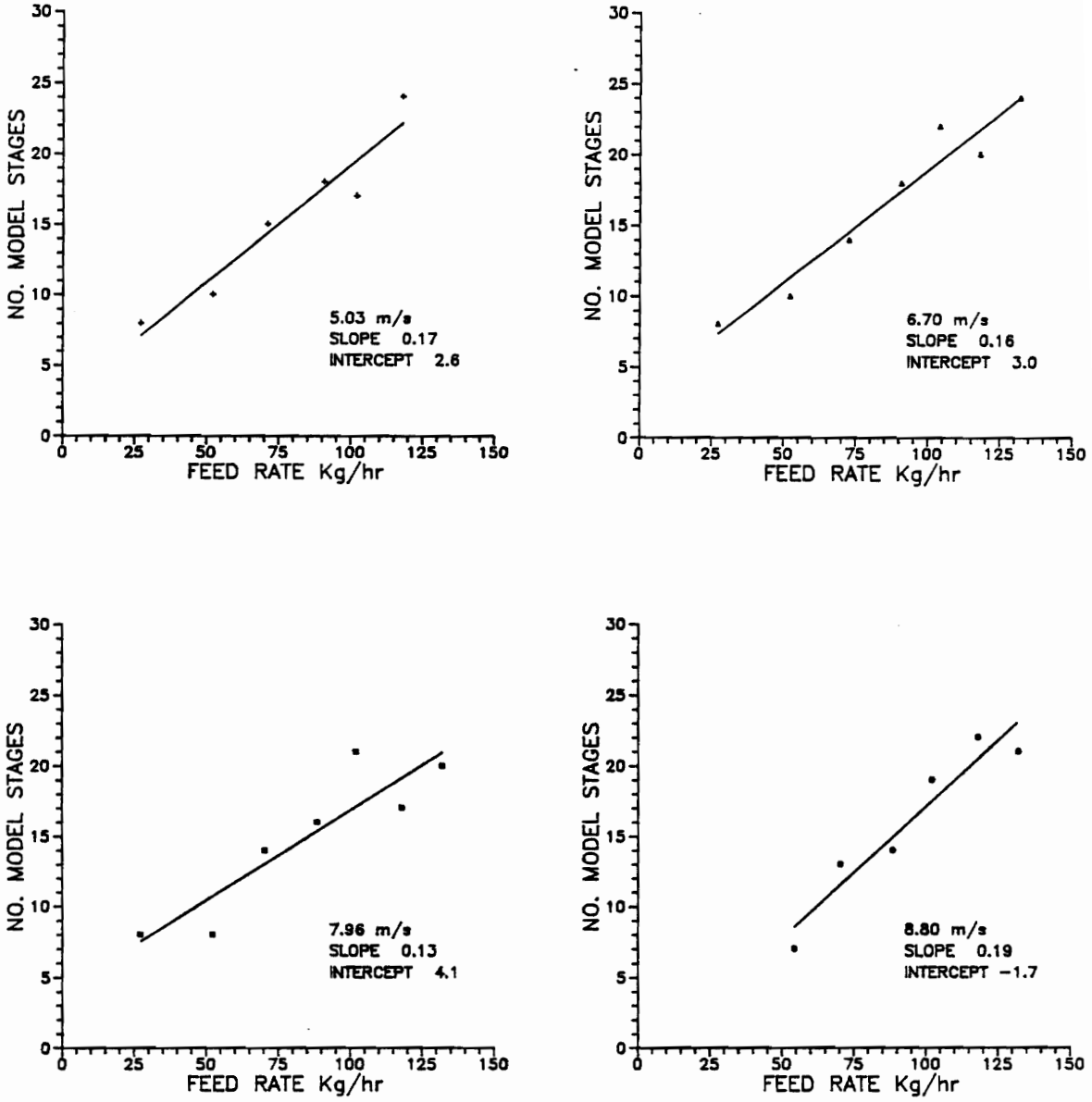


Figure 9.2. The relationship between the number of model stages and the processor feed rate for different rotor speeds.

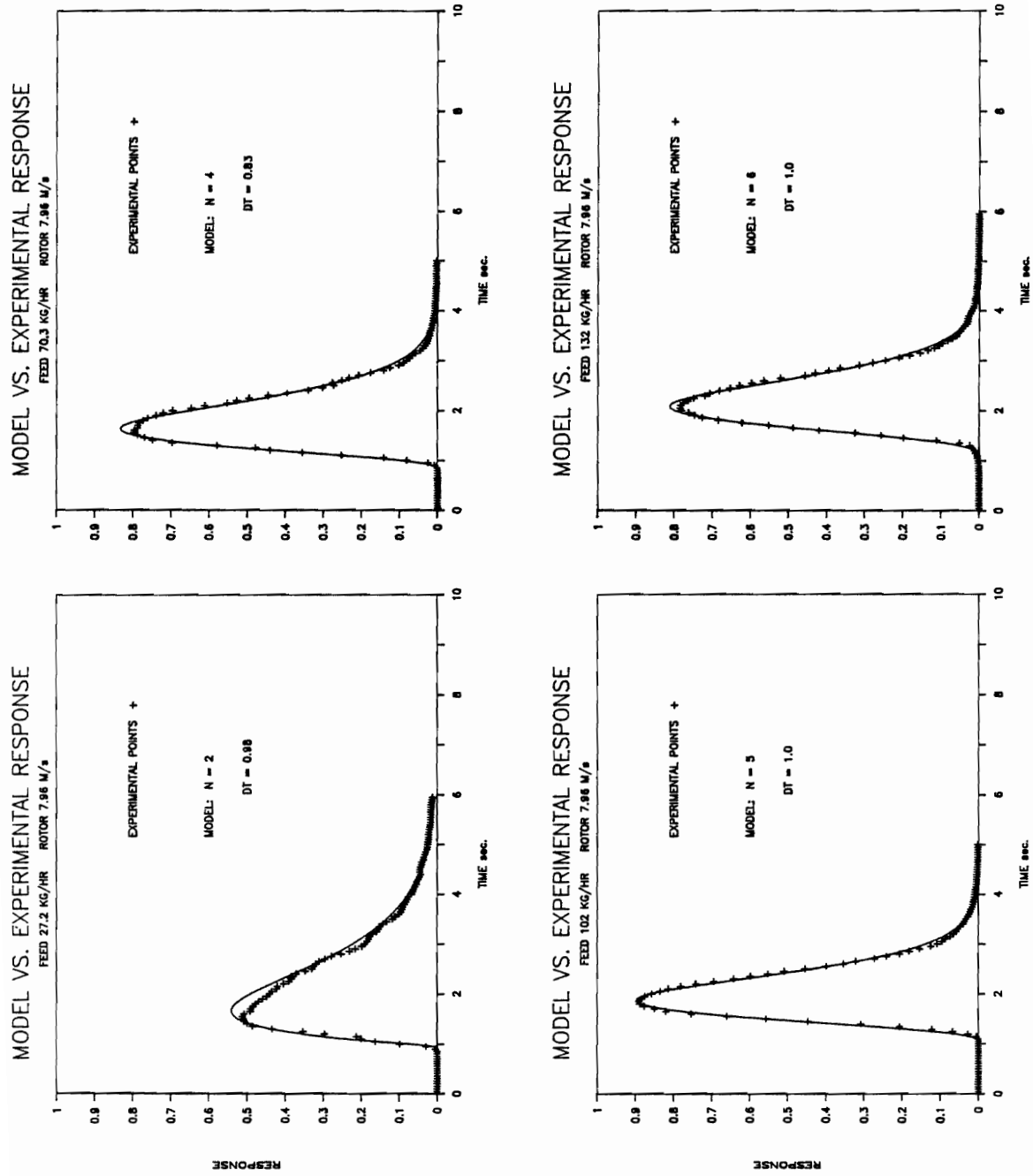


Figure 9.3. Fitting the tanks in series model with dead time 'Dt' to experimental data (at one rotor speed). The solid curves represent model predictions. Parameter 'N' is the number of ideal model stages. - A complete catalog of model fits is presented in appendix 2.

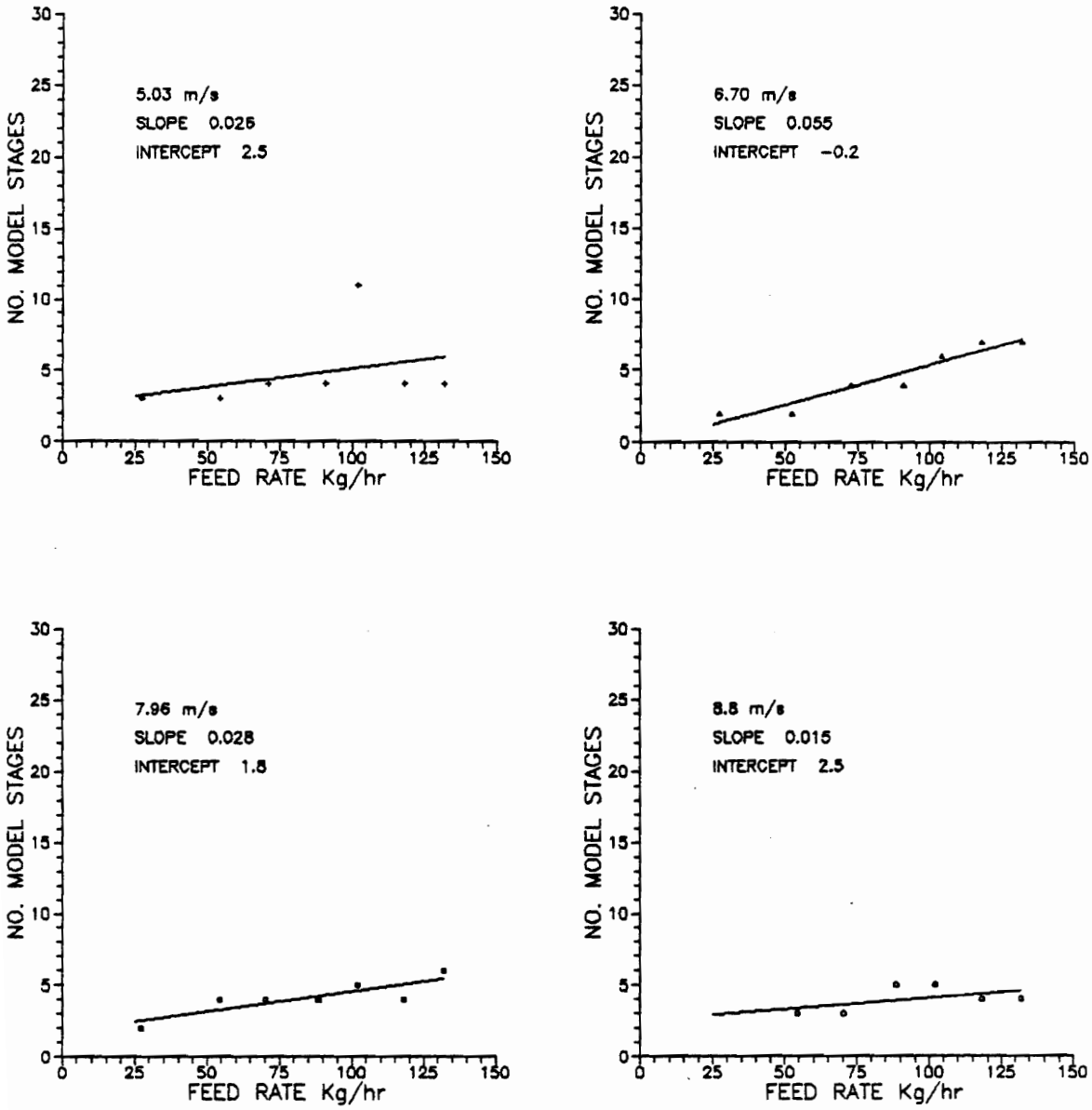


Figure 9.4. The relationship between the number of model stages - with dead time - and the feed rate for different rotor speeds.

by an ideal plug flow element at the beginning of the cascade. Alternately, Figure 9.5b distributes the dead time throughout the model, creating stages with both ideal plug flow and ideal stirred tank elements. Both models generate identical impulse responses / residence time distributions. Which is the more accurate physical representation of the process? Dead times on the order of those reported in Table 9.1 cannot be explained in terms of a lag introduced in the injection port (insensitivity of dead time to feed rate rules this out). The dead time is an integral characteristic of fluid convection inside the processor. The evidence suggests model 9.5b is the more physically correct.

The next question is: how can the dead time be incorporated into the staged cascade model coded into the numerical heat transfer simulation? A direct, if inelegant, translation of dead time can be done using the following stirred tank / plug flow identity, Levenspiel (1974): *As the number of stages N in a stirred cascade becomes large, the impulse response of the cascade approaches that of a plug flow element having the same total volume as the N stages.* Each of the dead time elements could be replaced by a battery of micro-stages pictured in Figure 9.5c. This straight forward approach, however, is flawed from a practical computing standpoint. Incorporation of a large number of stages, with small residence times, into the simulation results in an increase in calls to the B.V.P. solving routine. This increase is proportional to N^2. Computing costs, therefore, prevented the implementation of this model in the heat transfer simulation.

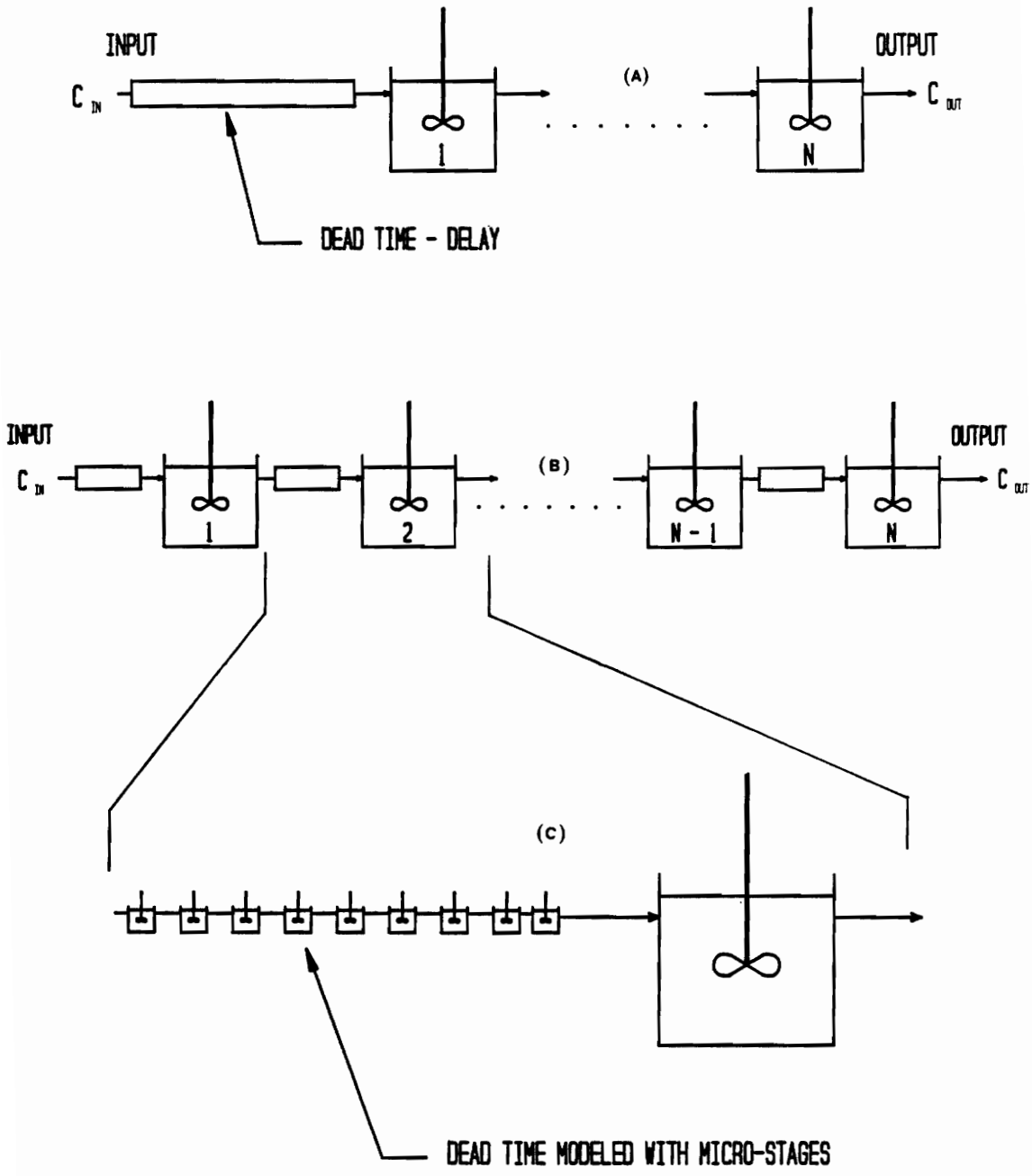


Figure 9.5. Equivalent representations of the model dead time: A) an initial consolidated dead time. B) an integrated dead time distributed among the stages. C) approximation of the dead time by a cascade of micro-stages.

The relationship between the feed rate and the resulting number of ideal mixed stages is shown for this model in Figure 9.4. These plots show a decreased dependence of the number of model stages on feed rate when compared to the model without dead time.

9.2 Quantitative Results From The Transient Heat Transfer Simulations

Intrinsic, or dispersion-corrected, process heat transfer film coefficients were generated using the transient heat transfer simulation program described in Chapter 6. The evaluations were carried out in accordance with the computation protocol of section 6.4; using the experimental transient temperature responses (plotted in Appendix 3) of the processor. For the purpose of comparison, film coefficients were also calculated from steady state experimental data using the log-mean temperature difference equation [2.1].

The results are given in Table 9.2 and Figure 9.6 where the process heat transfer film coefficients are plotted as a function of processor feed rate. Calculations based on the LMTD model assume no axial dispersion of fluid in the processor. A comparison of Figures 9.6 [top] and [bottom] indicates significant dispersion effects do obtain in the thin-film processor. Table 9.2 shows that these effects reduce the *apparent*, or LMTD, heat transfer film coefficient to 30 to 40% of the intrinsic value.

Axial dispersion of fluid always works to reduce axial temperature

gradients within a flowing system. This gradient relaxation results in a smaller average temperature difference between the heated wall and the bulk fluid inside the processor. The extreme case would be one of perfect back mixing where all the fluid in the processor is at the exit temperature. This results in the lowest possible ΔT, requiring the highest film coefficient to transport a given amount of energy. Thus, in a real processor, intrinsic film coefficients must always be greater than the calculated LMTD values.

TABLE 9.2

HEAT TRANSFER COEFFICIENT CALCULATIONS

LMTD CALCULATIONS FOR BASELINE RUNS					SIMULATION RESULTS	
FEED RATE kg/hr	LMTD deg C	NET Q Watts	U_o W m^{-2} K^{-1}	h_I	N	INTRINSIC COEF. h_I W m^{-2} K^{-1}
			ROTOR SPEED 5.03 m/s			
50					10	190
59.5	72.8	866	104	109		
88.1	64.5	1342	182	197	18	520
117	72.9	1297	156	167	24	670
132	64.2	1805	246	274		
			ROTOR SPEED 6.7 m/s			
50					10	220
59.5	71.5	866	106	111		
88.2	68.8	1210	154	165	18	530
117	72.1	1433	174	188	20	690
132	66	1750	232	257		
143					25	740
			ROTOR SPEED 7.96 m/s			
50					8	198
59.5	74.2	820	96.7	101		
88.2	65.8	1217	162	174	16	540
117	63.4	1658	229	253	17	585
132	63.4	1772	244	273		
143					20	770
			ROTOR SPEED 8.8 m/s			
59.5	69.3	847	107	112		
88.2	69.3	1232	155	166		
110	66.3	1592	210	230		
132	62.3	1878	264	297		

U_o represents the overall heat transfer coefficient; N the number of stages in the simulation. LMTD calculations are based on a heating side film coefficient of 8500 W/m^2-C.

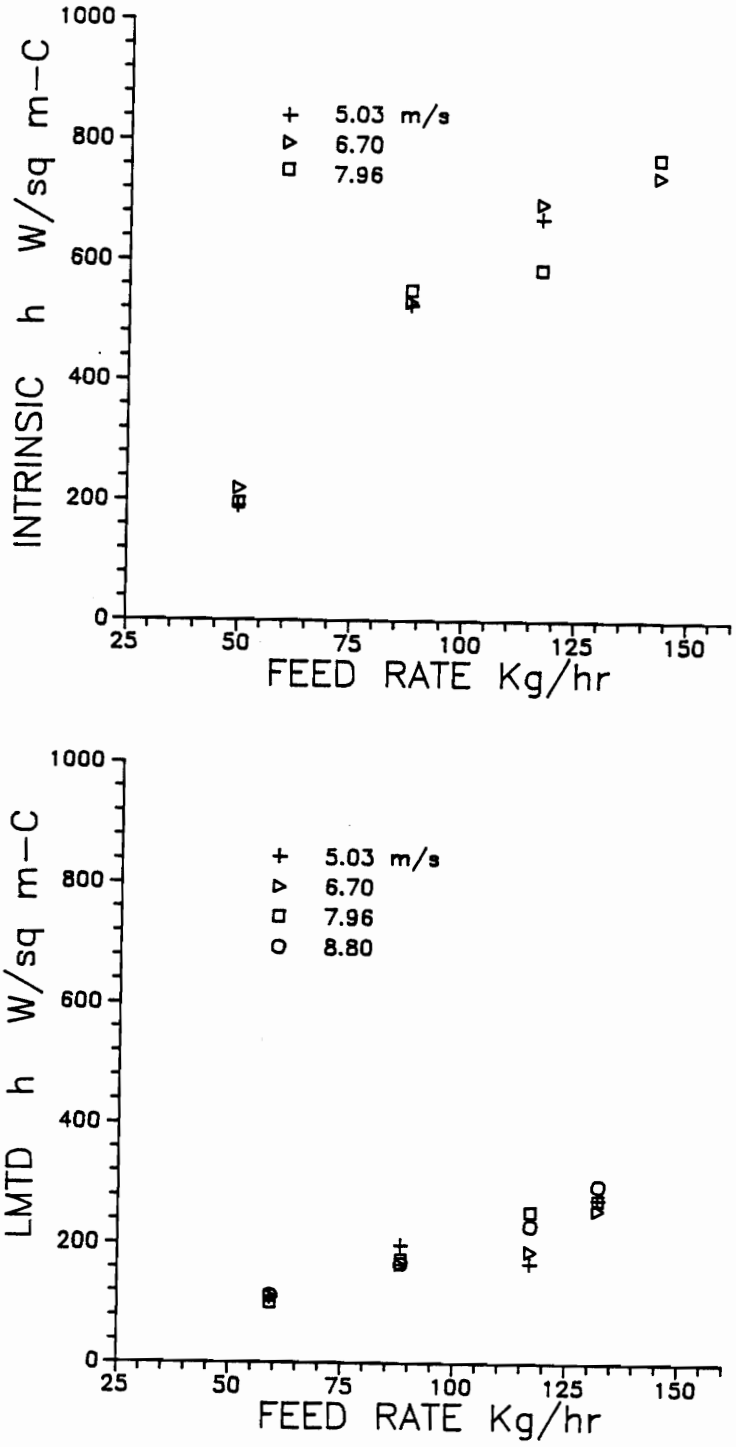


Figure 9.6. Experimental process heat transfer film coefficients calculated from: the tanks in series model using transient heat transfer data [top]; and the LMTD equation using steady state data [bottom].

9.3 Comparison Of Film Coefficients Obtained From The Transient Temperature Simulation With Those Obtained By Localized Flux Experiments

9.3.1 The Electrochemical Method

A number of investigators have developed experimental methods to measure *local*, or point, film coefficients (Bott and Azoory, 1969; Miyashita and Hoffman, 1978a and 1978b; Woschitz, 1983). In each case the object of the experiment was to generate and measure a small, highly localized, flux at the wall of an experimental device. These fluxes can be measured for a variety of operating conditions and are readily converted to film coefficients.

The most successful of these techniques has been the electrochemical method (ECM). In ECM experiments a conductive ionic solution is used as the process fluid. The ions are discharged at the surface of a small electrode embedded in the wall of the experimental device. The electrode current is measured and computations are made using the precisely known surface area of the electrode to yield a mass transfer coefficient. This mass transfer coefficient can be converted to a heat transfer film coefficient using an appropriate mass transfer / heat transfer analogy. The values obtained for heat transfer film coefficients by this method, however, are the values that would obtain under conditions of ideal plug flow. The method's inability to detect axial dispersion is a result of the isolated nature, and

small magnitude, of the electrode surfaces and their respective fluxes. The fluxes are not large enough to alter the bulk ion concentration in the process fluid. Dispersion, therefore, cannot relax an axial concentration gradient because none exists. Heat transfer coefficients obtained by the ECM will be uncorrected for dispersion. They represent a heat transfer regime where the average ΔT is considerably larger than the ΔT existing in a processor during actual operation, where the total unit surface contributes to the flux. This representation of excess ΔT by the electrochemical method predicts heat transfer film coefficients different from the intrinsic film coefficients calculated from transient heat transfer experiments. In order to properly compare these, the question becomes (i) can the ECM results have dispersion effects imposed upon them, or (ii) can the transient heat transfer results be extended to a dispersion free system? Fortunately, the answer to the second part is yes (with faith).

9.3.2 The Extended Cascade

As the number of ideal mixed stages in a cascade becomes large, the cascade approaches an ideal plug flow system. If transient temperature response experiments were carried out on such a system, the intrinsic heat transfer film coefficient would approach its plug flow value. It is interesting to speculate on what this value might be. That is, what heat transfer coefficient could be achieved in the thin-film processor if one could be designed to eliminate axial dispersion. The film coefficient must reach some

limiting value as the number of stages approaches infinity. A natural assumption is that the relationship between the film coefficient and the number of stages will be logarithmic (at least initially). This assumption was tested in Figure 9.7 with the existing experimental data. A linear fit superimposed on the data points corresponds to:

$$h = 552 \log_{10}(N) - 1016 \qquad [9.1]$$

Assume, for engineering purposes, that dispersion effects disappear somewhere in the range $N=10^2-10^3$. Under these conditions the heat transfer film coefficient might reach a value of 2000-3000 W/sq m-C from Figure 9.7. This is the value one would expect to find for a localized flux ECM experiment.

9.3.3 Comparison With Miyashita's Results

Extensive ECM experimental work has been done by Miyashita and Hoffman (1978b). Their laboratory apparatus was dimensionally similar to the processor used in this work (79 mm diameter, 0.127 mm rotor clearance) and their range of operating conditions was broad enough to allow comfortable extrapolation to the feed rates and rotor speeds of this work.

The results of Miyashita and Hoffman's ECM experiments are presented in the bottom plot of Figure 9.8. The upper plot is a reproduction of the correlation in Figure 9.7, with the axes extended. Comparing the two, it

can be seen that for rotor speeds from 5-9 m/s, Miyashita and Hoffman's work would predict film coefficients of 2500-3500 W m^{-2} K^{-1}. While the tentative nature of these extrapolations should be stressed, the numbers that result are very consistent.

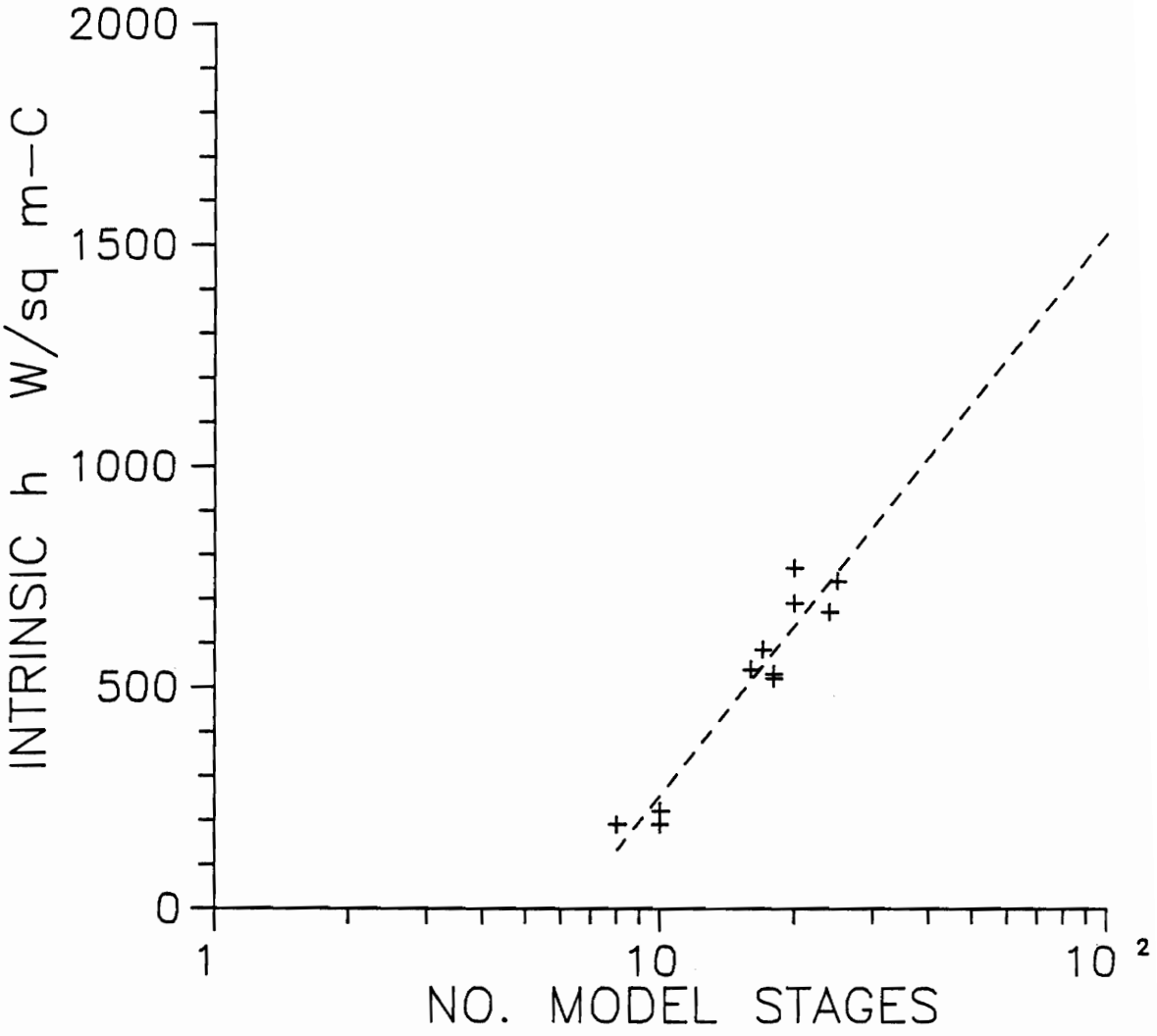


Figure 9.7. A qualitative test of the logarithmic relationship between the intrinsic heat transfer film coefficient and the number of model stages.

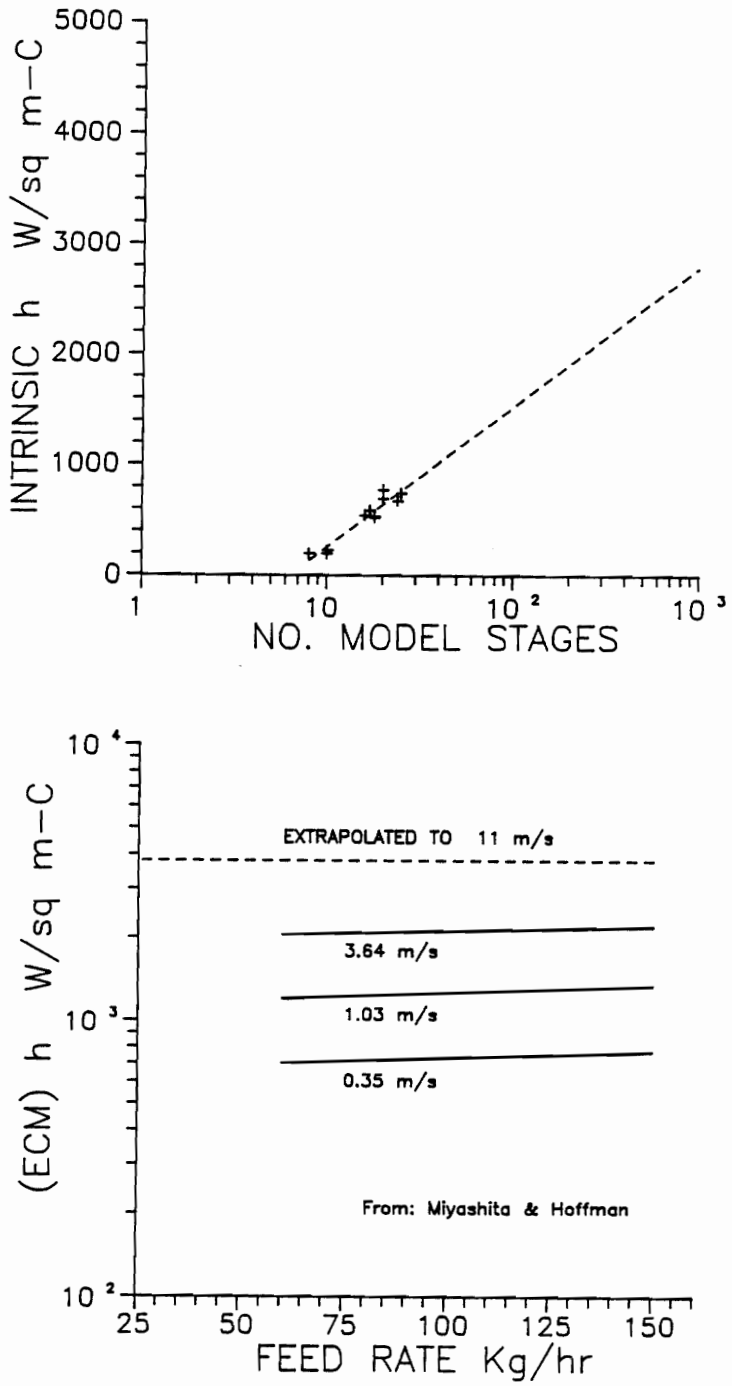


Figure 9.8. A comparison of intrinsic film coefficients determined by this work and extrapolated to high values of 'N' [top], with film coefficients determined by Miyashita and Hoffman using the electrochemical method [bottom] in a dimensionally similar apparatus. - ref. discussion in sections 9.3.2-3.

CHAPTER TEN

Conclusions and Recommendations

10.1 Assessment of the Penetration Theory

As early as Harriot (1959), investigators have reported that penetration theory over-predicts the value of heat transfer film coefficients calculated from experimental data by the LMTD method. To explain this fact, Trommelen et al. (1971) suggested that radial mixing may be inhibited in scraped-surface processors, thus reducing the heat transferred by thermal penetration. It is reasonable to suppose that some of the hot fluid elements swept from the heat transfer surface will be re-deposited on that surface by centrifugal force before giving up their energy to the bulk fluid. Other workers have submitted axial dispersion as the primary mechanism depressing heat transfer rates in scraped-surface equipment below those predicted by penetration theory (Bott, Azoory and Porter,1968; Maingonnet and Corrieu, 1986b).

The many important differences between scraped-surface and thin-film processors not withstanding, these two mechanisms are probably operative in both types. Figure 10.1 can be used to examine the magnitude of each effect. The disparity between the calculated (LMTD), and the predicted (penetration theory), heat transfer film coefficients is assumed to result from the combined effects of axial dispersion and incomplete radial mixing. The

intrinsic film coefficients are corrected for axial dispersion but not for incomplete radial mixing. Therefore, the magnitude of the difference between the intrinsic coefficients and the penetration coefficients suggests that the hot fluid elements are short circuiting back to the heat transfer wall. While this explanation may seem arbitrary, it can be rationalized by the following argument: *The cascade model provides an unambiguous correction for axial dispersion.* That the model does *not* correct for incomplete radial mixing is less obvious. Recall, however, the model parameters are evaluated from residence time distributions. These are simply probability distributions for the (average) axial velocity of a fluid element. The distributions contain no information about the radial component of the velocity field. The cascade model that results has no provision for 'radial' gradients and, therefore, perfect radial mixing obtains in the model by default.

Figure 10.1 suggests some conclusions concerning macroscopic convection in the thin-film processor:

1. Under the conditions of this study, the penetration theory over predicts heat transfer coefficients in the thin-film processor by an order of magnitude.
2. Axial dispersion of the process fluid plays an important role in the overall heat transfer characteristics of the processor.
3. Radial re-deposition of hot fluid elements on the heat transfer wall may significantly lower the driving force for heat transfer in the processor.

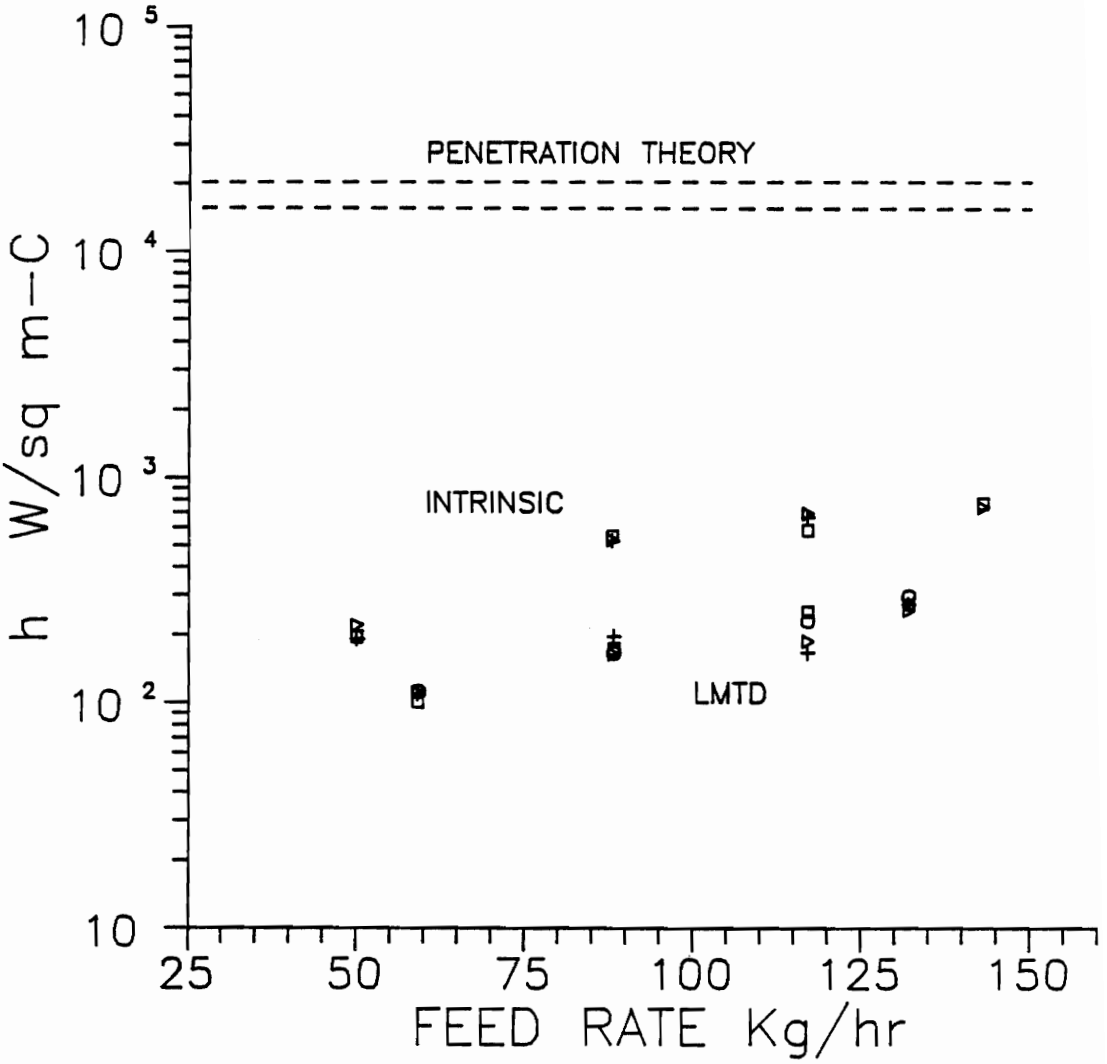


Figure 10.1. Comparison of the film coefficients calculated from experimental data, using the transient simulation and LMTD methods, with those predicted by the penetration theory. Note penetration theory over predicts heat transfer film coefficients in the thin-film processor by two orders of magnitude.

While penetration theory has enjoyed reasonable success in predicting heat transfer film coefficients in scraped-surface heat exchangers (Azoory and Bott, 1970), its failure with thin-film equipment is not really surprising. The penetration film coefficient of equation [1.7] increases as the square root of rotor frequency. At the low circumferential speeds characteristic of zero clearance scraped-surface processors (0.3 m/s Bott, et al., 1968a and 1968b) reasonable predictions may be obtained. At the high speeds of fixed clearance thin-film processors, the penetration film coefficient equation behaves as an unbounded function of rotor frequency. A limitation is certainly reached at some point, but the question is whether these calculated penetration coefficients, which are 1-2 orders of magnitude greater than the experimental values, have any physical meaning.

10.2 Success Of The Linear Model, And Model Based Numerical Simulation

This work has demonstrated the ability of simple deterministic linear models to quantify the state of axial fluid convection inside an operational thin-film processor. It has also demonstrated the feasibility of using a lumped parameter model as the basis for numerical simulation of an operational thin-film processor. The significance of these findings is:

1. They provide a complementary alternative to the current academic approaches for studying heat transfer in thin-film processors.
2. The experimental methods and/or simulation results can be directly

applied to operational processors.

3. These techniques can be used to predict the transient behavior of thin-film equipment.

The drawbacks to the approach are that transient heat transfer experiments are more difficult than steady state experiments, and the simulation is computationally expensive.

10.3 Application Of The 'Intrinsic' Heat Transfer Film Coefficient

In Chapter One, the intrinsic, film, heat transfer coefficient was proposed as a generalizable parameter for the design and scale up of thin-film (sensible) heat transfer operations. However, the experimental results only partially support this idea. If dispersion effects alone suppress heat transfer in thin-film processors, then the intrinsic film coefficients obtained from a pilot scale experiment could be applied to a large unit through the large unit's unique model simulation. In principle, the sensible heat transfer performance (for a given process fluid) of any subsequent unit could be determined via its model simulation. However, experimental evidence indicates incomplete radial mixing may contribute to the suppression of heat transfer as well. Although this effect is difficult to quantify, it cannot be neglected for the present.

10.4 Suggestions For Further Work

The first priority for continued work is actual verification of the scale up capability of this method. Such a project would involve performing tracer injection experiments on a larger unit in the Luwa Corporation test center. The variables in the processor simulation program would be changed to match the characteristics of the new unit. The program would then be run over a range of operating conditions using the intrinsic heat transfer film coefficients given in Figure 10.1. Finally, the actual unit would be operated at these points and the experimental and simulated results compared. If they are in good agreement, the intrinsic heat transfer film coefficient will have been shown to be an effective scale up parameter for thin-film equipment operating in the sensible heating mode.

The predictive capability of the simulation can also provide information about the effect of incomplete radial mixing, or re-deposition of the hot fluid elements on the heat transfer wall. If the intrinsic film coefficients determined for one processor satisfactorily predict the heat transfer performance of a different processor, then one of the following statements applies to the phenomena of radial re-deposition:

1. The relative magnitude of the re-deposition effect is negligible in comparison to the effects of axial dispersion on heat transfer performance.

2. Re-deposition effects are significant, but their scale up is directly proportional to the effects of axial dispersion, and the unit scale up is not affected.

The second statement is improbable although it cannot be discounted out of hand.

Conversely, if the predictive capability of the simulation is *not* satisfactory, i.e. the intrinsic film coefficient does *not* remain constant from model to model at similar processing conditions, then the discrepancy should be the result of differences in radial mixing between the two processors (recalling that the models explicitly include the effects of axial dispersion). Therefore, if errors in the scale up prediction exist, they provide a direct probe of the effect of radial fluid stagnation and re-deposition within the thin-film processor.

Another direction for future work is the incorporation of the, more physically correct, dead time fluid convection model into the numerical heat transfer simulation. The fact that dead time remained constant in the experimental processor (at about 1 s) over a wide range of operating conditions suggests that it is a fundamental property of the processor rather than a flowrate-dependent effect. In Figure 9.5 the dead time is represented as a plug flow element(s). A key question is, should this plug flow element participate in the transfer of heat? An answer can be obtained by re-running the simulation, per section 6.4, using the number of stages

(N) called for by the dead time model, and delaying the simulation output by the value of the dead time (Dt). If the model-to-data fit improves, the dead time can be safely modeled as a simple delay. In this case the simulation program will not require re coding to incorporate the dead time model.

10.5 Archived Data

All of the data plotted in the appendix has been archived on 5 1/4 inch DS/DD floppy disks. This data is stored in a Lotus 123 (R), version 2, worksheet format. All post-run manipulation of the experimental data was done in these worksheets and documented with comments. Typical operations include the normalization of impulse response curves and the synchronization of starting times for the transient temperature curves...etc. The raw experimental data logged by Labtech Notebook (R) has also been archived on disks. The total archive (approximately 50 disks) is available for inspection and copying from Kim Hunter through the Chemical Engineering Department at Virginia Tech.

LITERATURE CITED

Asbjornsen, O. A., "The Distribution of Residence Times in a Falling Film", Chem. Eng. Sci. **14**, 211-27 (1961).

Azoory, S. and Bott, T. R., "Local Heat Transfer Coefficients in a Model 'Falling Film' Scraped Surface Exchanger", Can. J. Chem. Eng. **48**, 373-77 (1970).

Berg, P. and McGregor, J., "Elementary Partial Differential Equations", Holden-Day, Oakland, Ca. (1966).

Bird, R., Stewart, W., Lightfoot, E., "Transport Phenomena", Wiley, N.Y. (1960).

Bott, T. R., Azoory, S., Porter, K. E., "Scraped-Surface Heat Exchangers Part I - Holdup and Residence Time Studies", Trans. Instn. Chem. Engrs. **46**, T33-36 (1968a).

Bott, T. R., Azoory S., Porter, K. E., "Scraped-Surface Heat Exchangers Part II - the Effects of Axial Dispersion on Heat Transfer", Trans. Instn. Chem. Engrs. **46**, T37-43 (1968b).

Bott, T. R. and Azoory, S., "Heat Transfer in Scraped Heat Exchangers", Chem. Proc. Eng., 85-90 January (1969).

Boyce, W. E. and DiPrima, R. C., "Elementary Differential Equations and Boundary Value Problems", Wiley, N. Y. (1987).

Calu, M. P. and Lameloise, M. L., "The Interpretation of Residence-Time Distribution Measurements in Flows of Variable Density. Applications in the Modeling of Climbing-Film Evaporators for the Sugar Industry", Int. Chem. Eng. **28**, 424-434 (1988).

Castellan, G. W., "Physical Chemistry", Addison-Wesley, Reading, MA (1971).

Clements, W. C. and Schnell, K. B., "Pulse Testing For Dynamic Analysis", Ind. Eng. Chem. Design and Development **2**, No. 12, 92-102 (1963).

Clements, W. C., "A Note on Determination of the Parameters of the Longitudinal Dispersion Model From Experimental Data", Chem. Eng. Sci. **24**, 957 (1969).

Cramers, H. and Alberda, G., "Frequency Response Analysis of Continuous Flow Systems", Chem. Eng. Sci. 2, 173-181 (1953).

Davis, M. E., "Numerical Methods and Modeling For Chemical Engineers", Wiley, N. Y., p. 128 (1984).

Dessy, R. E., "Laboratory Automation", Class Notes For Chem. 5010, Copyrighted by the American Chemical Society, Washington D. C., Published by Virginia Polytechnic Institute, Blacksburg, VA, pp. 157-167 (1978).

Duchateau, P. and Zachmann, D., Schaum's Outline Series, "Theory And Problems of Partial Differential Equations", McGraw-Hill, N.Y. (1986).

Glover, William B., Vice President of Engineering, Luwa Corporation, personal correspondence (1987).

Harriot, P., "Heat Transfer in Scraped-Surface Heat Exchangers", Chem. Eng. Progr. Symp. Ser. 55 (29), 137–139 (1959).

Holman, J. P., "Heat Transfer", McGraw-Hill, N. Y. (1972).

Hopkins, M. J., Sheppard, A. J. and Eisenklam, "The Use Of Transfer Functions in Evaluating Residence Time Distribution Curves", Chem. Eng. Sci. 2, 1131-1137 (1969).

Hougen, J. O. and Walsh, R. A., "Pulse Testing Method", Chem. Eng. Prog. 57, 69-79 (1961).

Kern, D. Q., "Process Heat Transfer", McGraw–Hill, N. Y. (1950).

Kern, D. Q. and Karakas, H. J., "Mechanically Aided Heat Transfer", Chem. Eng. Progr. Symp. Ser. 55 (29), 141-148 (1959).

Khang, S. J. and Fitzgerald, T., "A New Probe And Circuit For Measuring Electrolyte Conductivity", Ind. Eng. Chem. Fund., 14, 208-213 (1975).

Kool, J., "Heat Transfer in Scraped Vessels and Pipes Handling Viscous Materials", Trans. Inst. Chem. Engrs., 36, 253-258 (1958).

Kuester, J. L. and Mize, J. H., "Optimization Techniques With Fortran", McGraw-Hill, N. Y. (1973).

"Labtech Notebook Manual", Version 2.5 (Metrabyte(R)), Laboratory Technologies Corp., Wilmington Ma. (1984).

Lapidus, L. and Seinfeld, J. H., "Mathematical Methods in Chemical Engineering Vol. 3 Process Modeling, Estimation, and Identification", Chap. 6-7, Prentice-Hall, N. J. (1974).

Levenspiel, O., "Chemical Reaction Engineering 2nd. Ed.", Chap. 9, Wiley, N.Y. (1972).

Maingonnet, J. F. and Corrieu, G., "A Study of the Thermal Performance of a Scraped-Surface Heat Exchanger. Part I - Review of The Principal Models Describing Heat Transfer and Power Consumption", Int. Chem. Eng. **26**, No. 1, 44-54 (1986a).

Maingonnet, J. F. and Corrieu, G., "A Study of the Thermal Performance of a Scraped-Surface Heat Exchanger. Part II - The Effect of the Axial Diffusion of Heat", Int. Chem. Eng. **26**, No. 1, 55-68 (1986 b).

McClamroch, N. H., "State Models of Dynamic Systems", Chap. 1-3, Springer-Verlag, N.Y. (1980).

Michelsen, M. L. and Ostergaard, K., "The Use of Residence Time Distribution Data For Estimation of Parameters In Axial Dispersion Model", Chem. Eng. Sci. **25**, 583-592 (1970).

Miyashita, H. and Hoffman T. W., "Local Heat Transfer Coefficients in Wiped-Film Heat Exchangers", Heat Transfer 1978, Vol. 4, pp. 1-6 (1978a).

Miyashita, H. and Hoffman T. W., "Local Heat Transfer Coefficients in Scraped-Film Heat Exchanger", J. Chem. Eng. Japan **11**, 444-50 (1978b).

Myers, R. H., "Response Surface Methodology", Edwards Brothers Inc. (1976).

Ostergaard, K. and Michelsen, M. L., "The Use of the Imperfect Tracer Pulse Method For Determination of Hold-up and Axial Mixing", Can. J. Chem. Eng. **47**, 107-112 (1969).

Penney, W. R. and Bell, K. J., "Close-Clearance Agitators. Part 2 - Heat Transfer Coefficients", Ind. Eng. Chem. **59**, No. 4, 47-54, April (1967).

Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., "Numerical Recipes the Art of Scientific Computing", Cambridge University Press, N. Y. (1986).

Ramirez, R. W., "The FFT Fundamentals and Concepts", Prentice-Hall, N. J., pp. 66-75 (1985).

Thomas, L., "An Analysis of Heat Transfer for Turbulent Axial Flow in Close Clearance Exchangers", Chem. Eng. Sci. 27, 1029-34 (1972).

Trommelen, A. M., Beek, W. J., Van De Westelaken, H. C., "A Mechanism for Heat Transfer in a Votator-Type Scraped-Surface Heat Exchanger", Chem. Eng. Sci. 26, 1987-2001 (1971).

"Users Manual IMSL Library", Vol. 1, Ch. D, Edition 9.2 (1984).

Watanabe, M., Toyama, M., Nkamura, K., "Fundamental Investigation of the Flow in an Agitated Thin Film Evaporator", Bul. JSME 19, No.135, September (1976).

Woschitz, D., "Heat Transfer in Mechanically Agitated Thin-Film Equipment", Tech. Film Appl. Oper. Genie Chim., Colloq. Int., Inst. Fr. Pet., Rueil-Malmaison, France, pp. 276-316 (1983).

APPENDIX 1

RESIDENCE TIME DISTRIBUTION PLOTS

Note: 'TAU' represents the mean residence time.

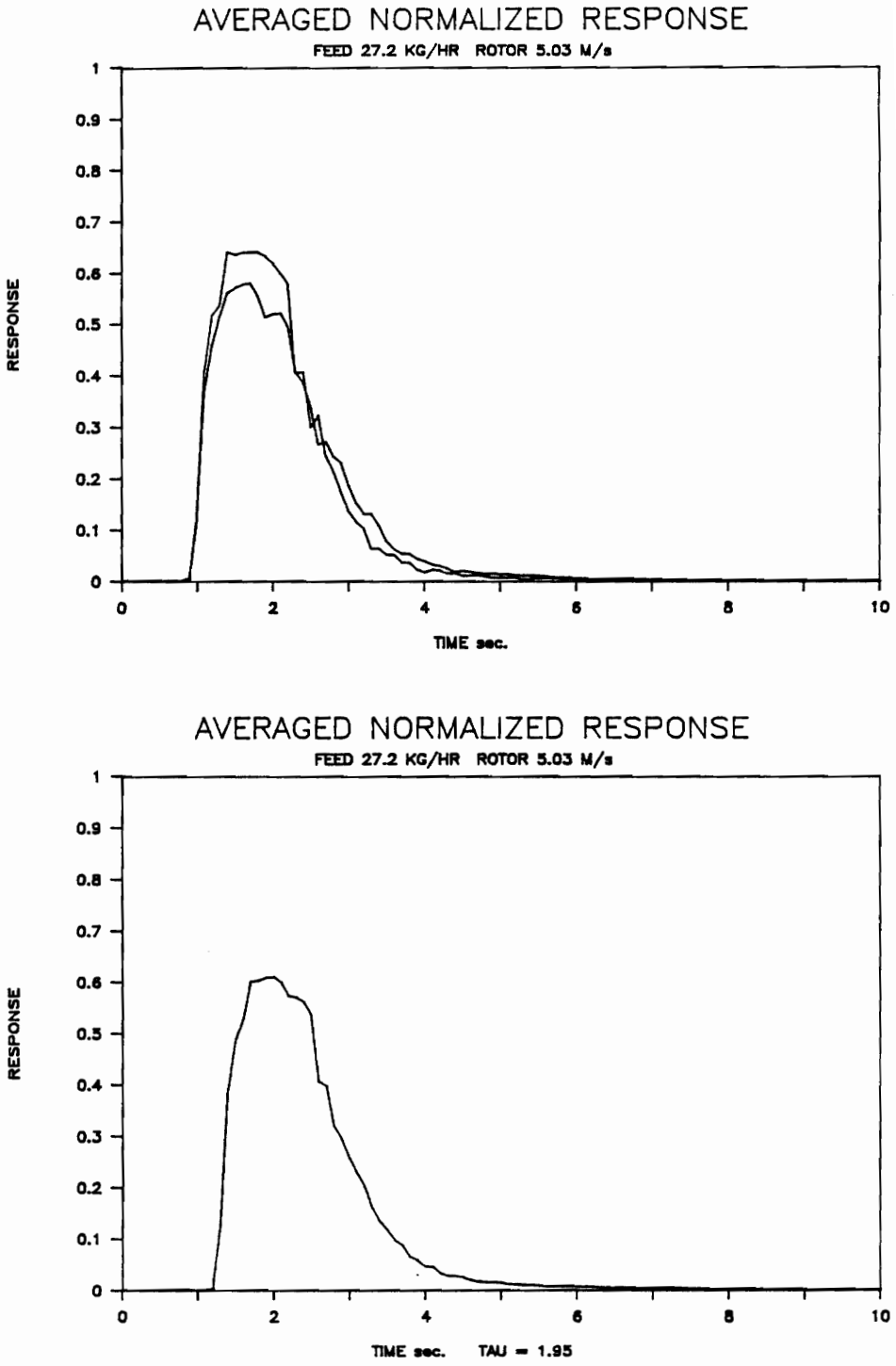


Figure A.1.1. Individual and averaged residence time distributions. TAU represents the mean residence time.

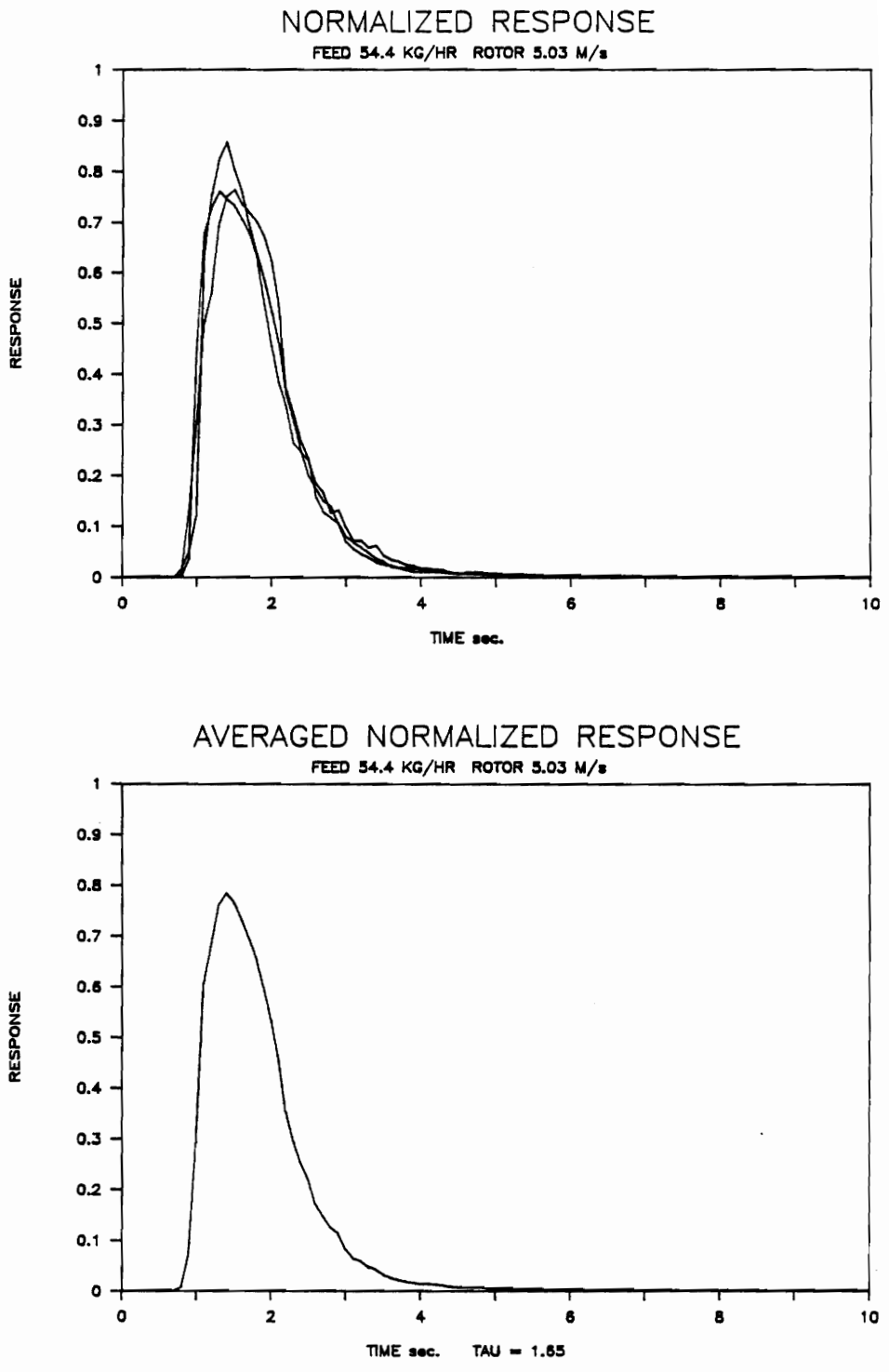


Figure A.1.2. Individual and averaged residence time distributions. TAU represents the mean residence time.

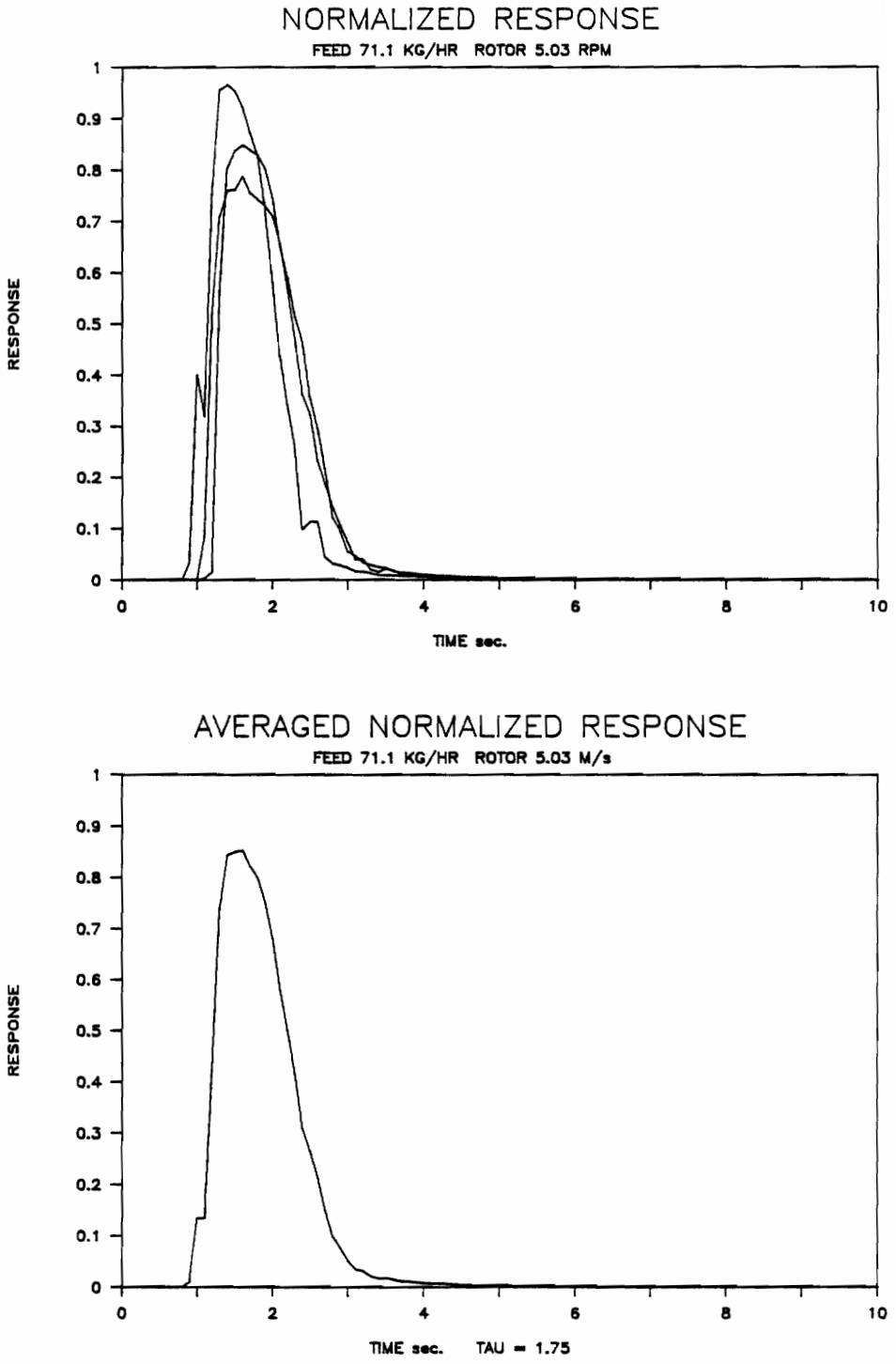


Figure A.1.3. Individual and averaged residence time distributions. TAU represents the mean residence time.

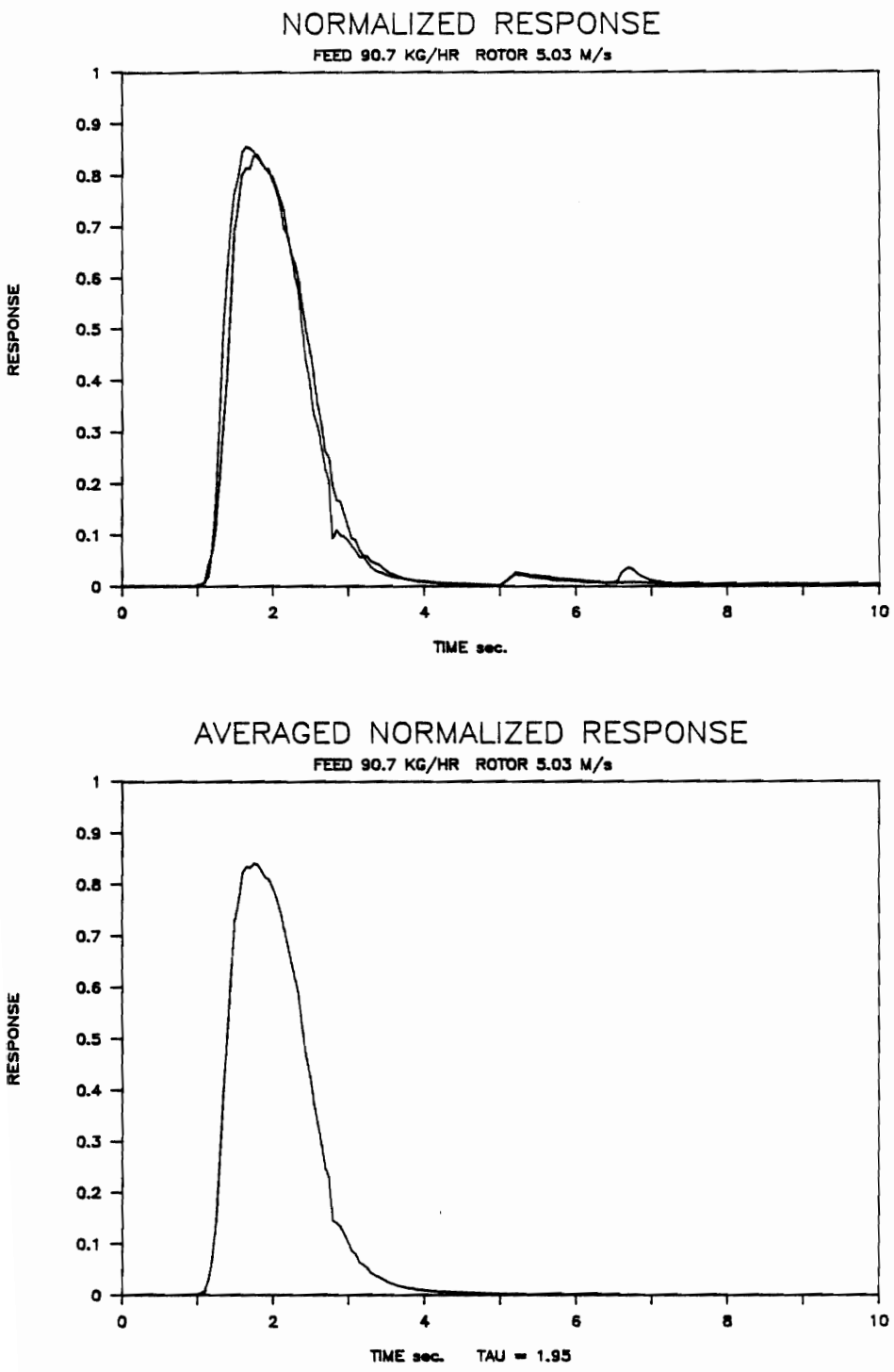
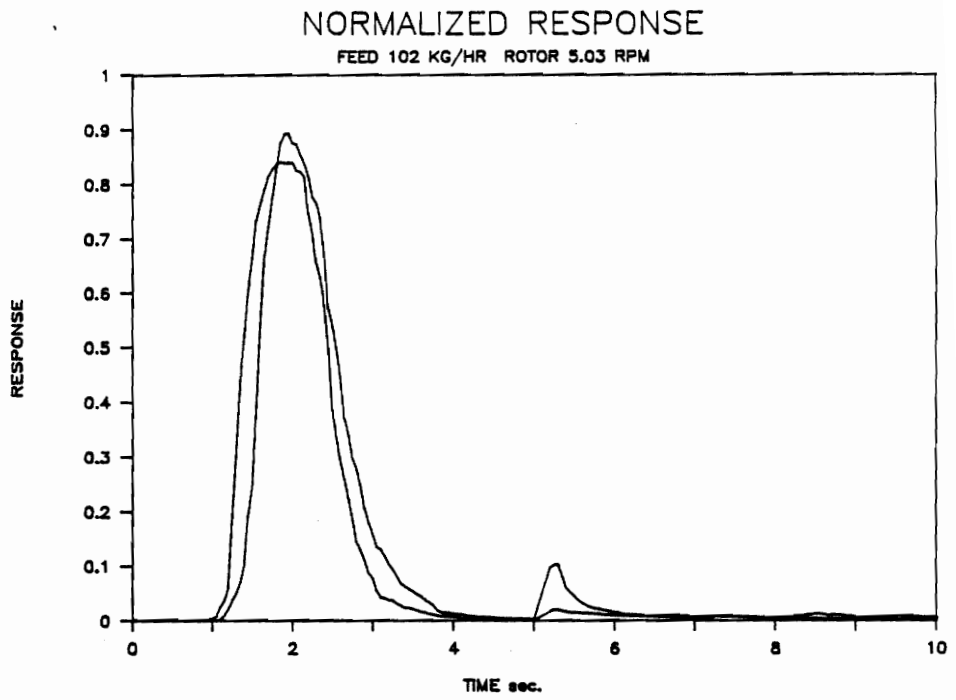


Figure A.1.4. Individual and averaged residence time distributions. TAU represents the mean residence time.



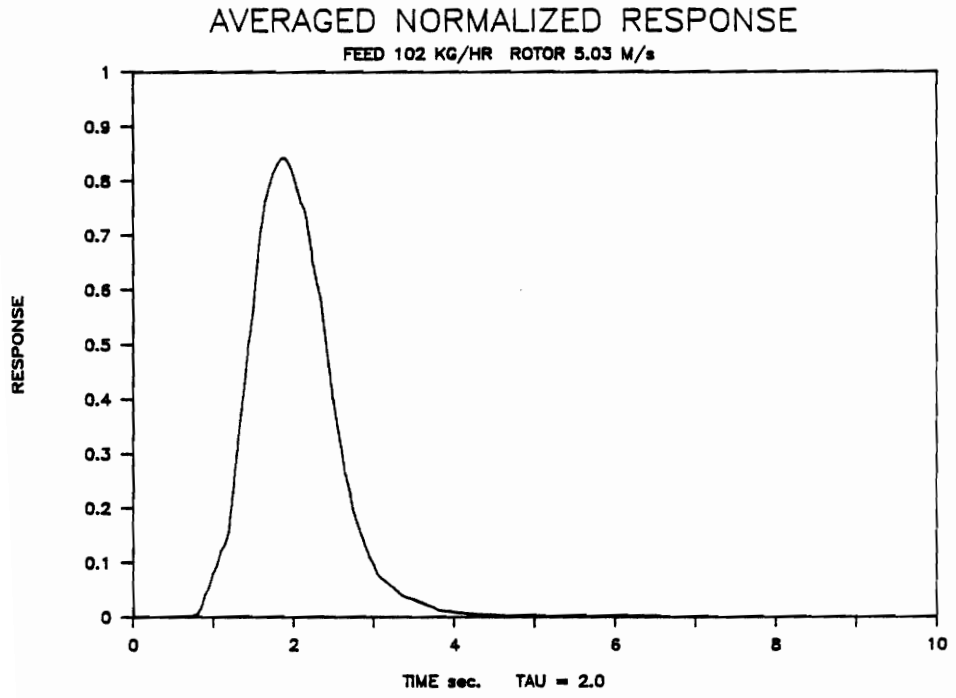
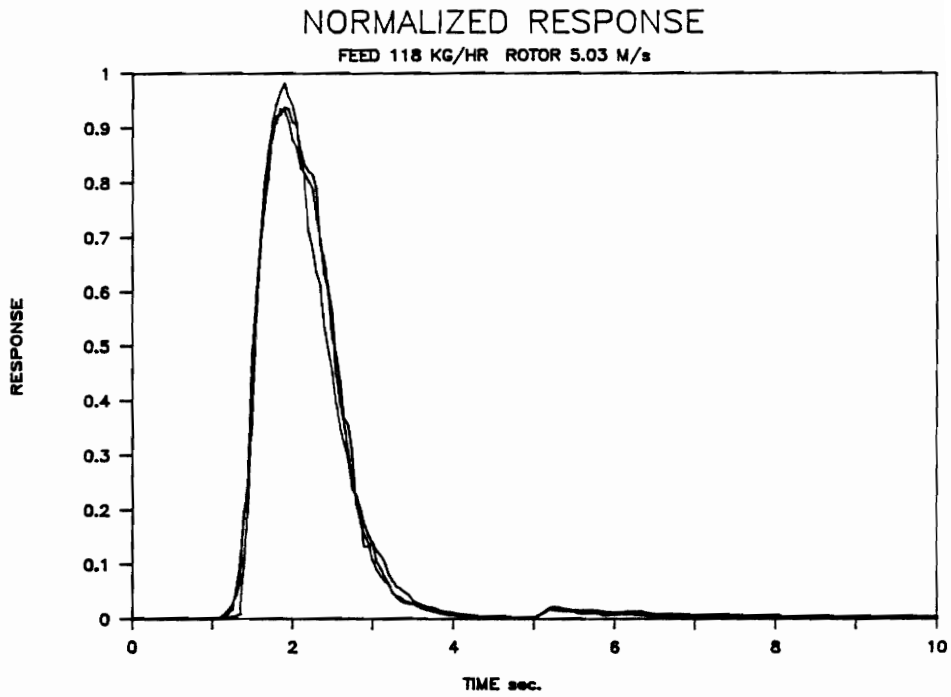


Figure A.1.5. Individual and averaged residence time distributions. TAU represents the mean residence time.



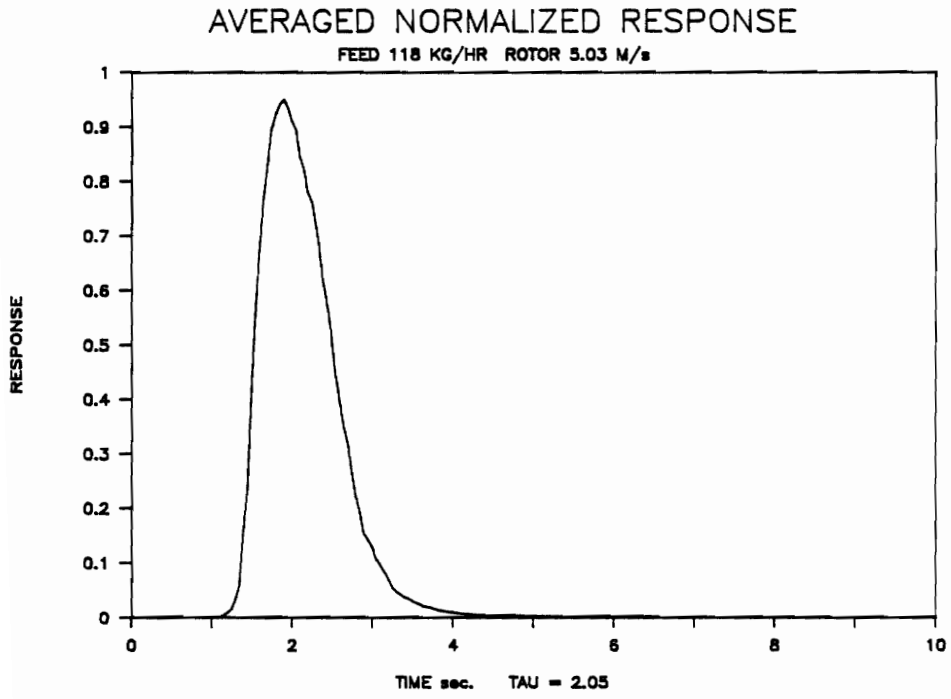


Figure A.1.6. Individual and averaged residence time distributions. TAU represents the mean residence time.

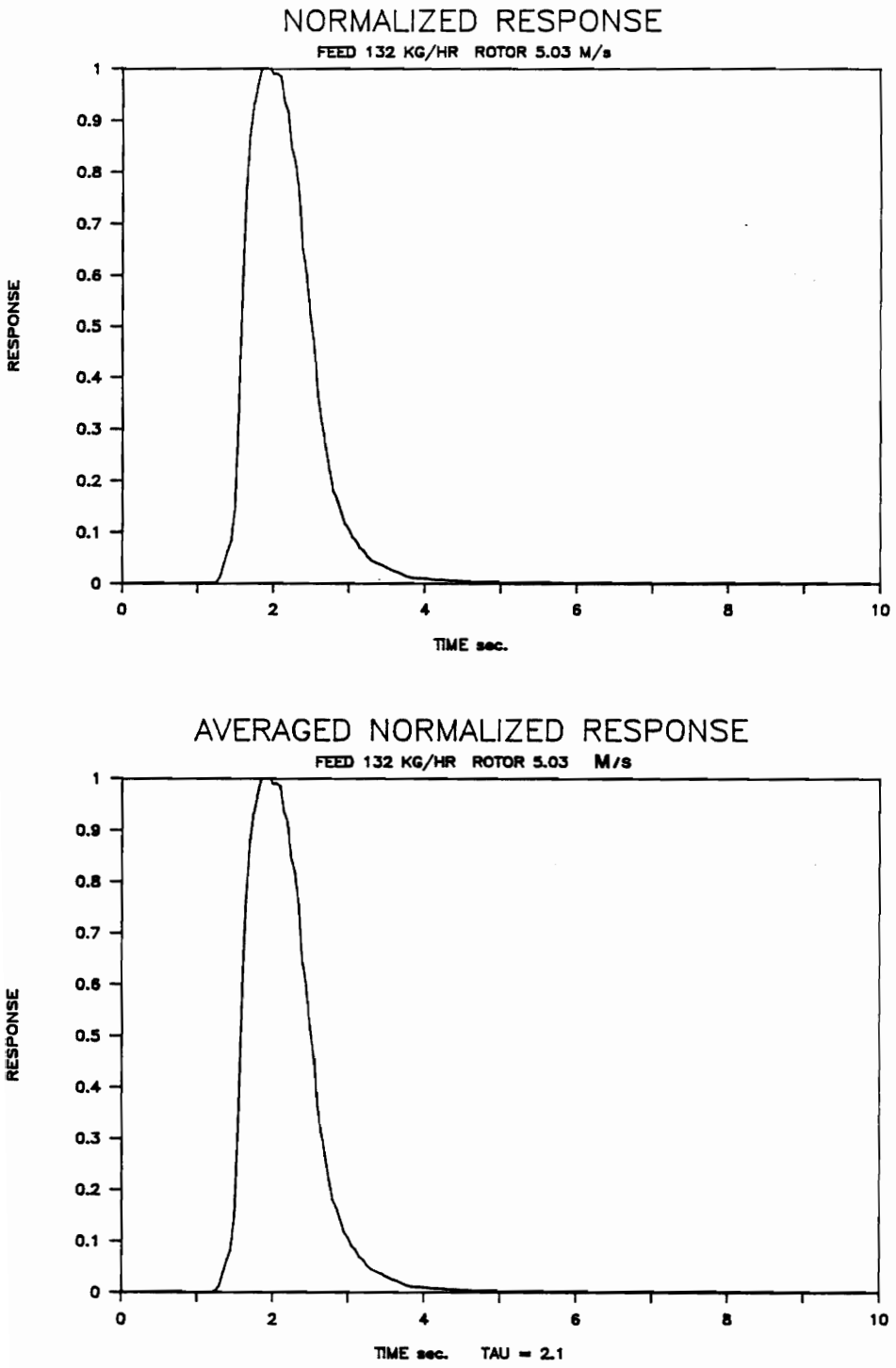


Figure A.1.7. Individual and averaged residence time distributions. TAU represents the mean residence time.

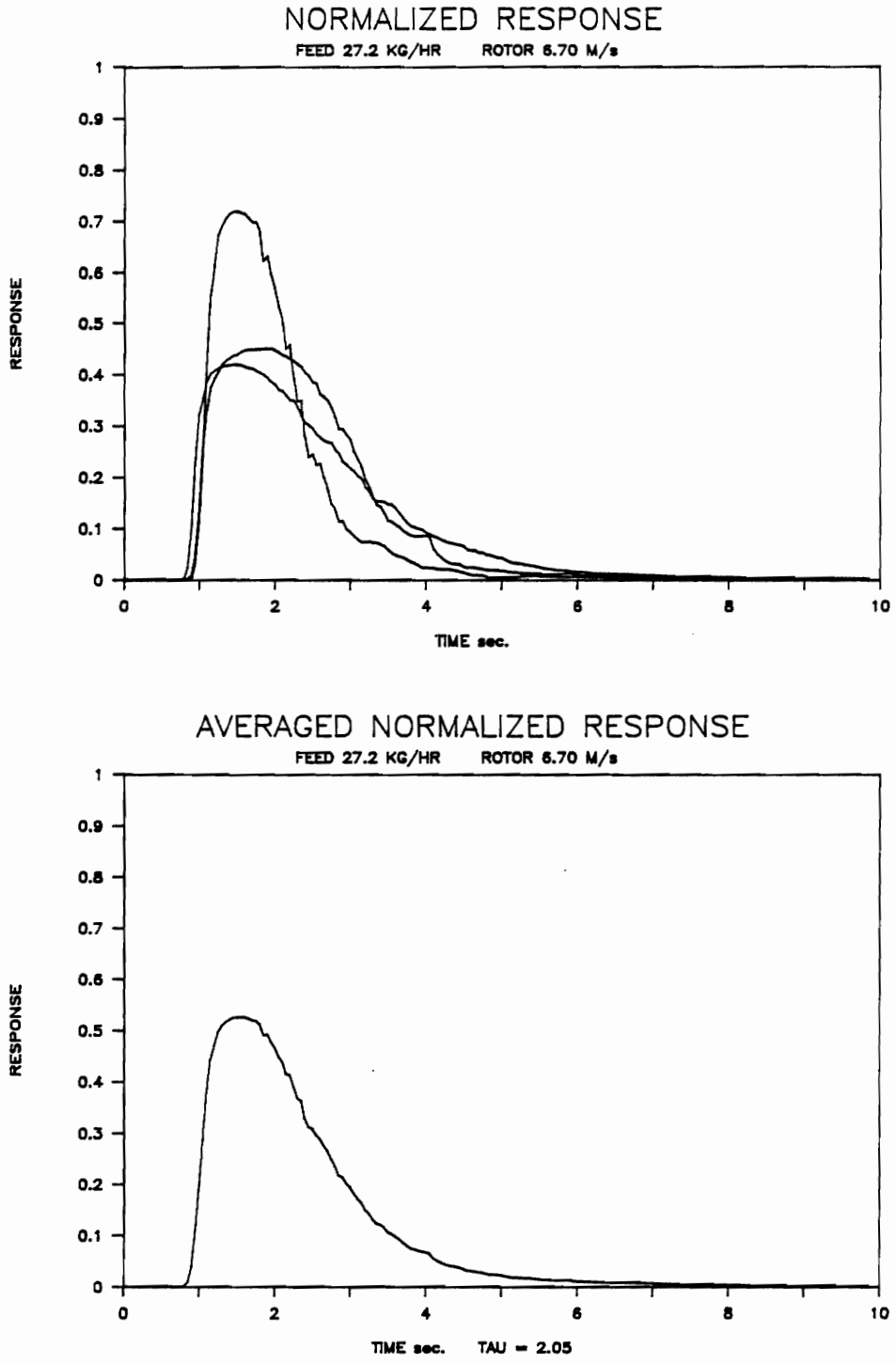


Figure A.1.8. Individual and averaged residence time distributions. TAU represents the mean residence time.

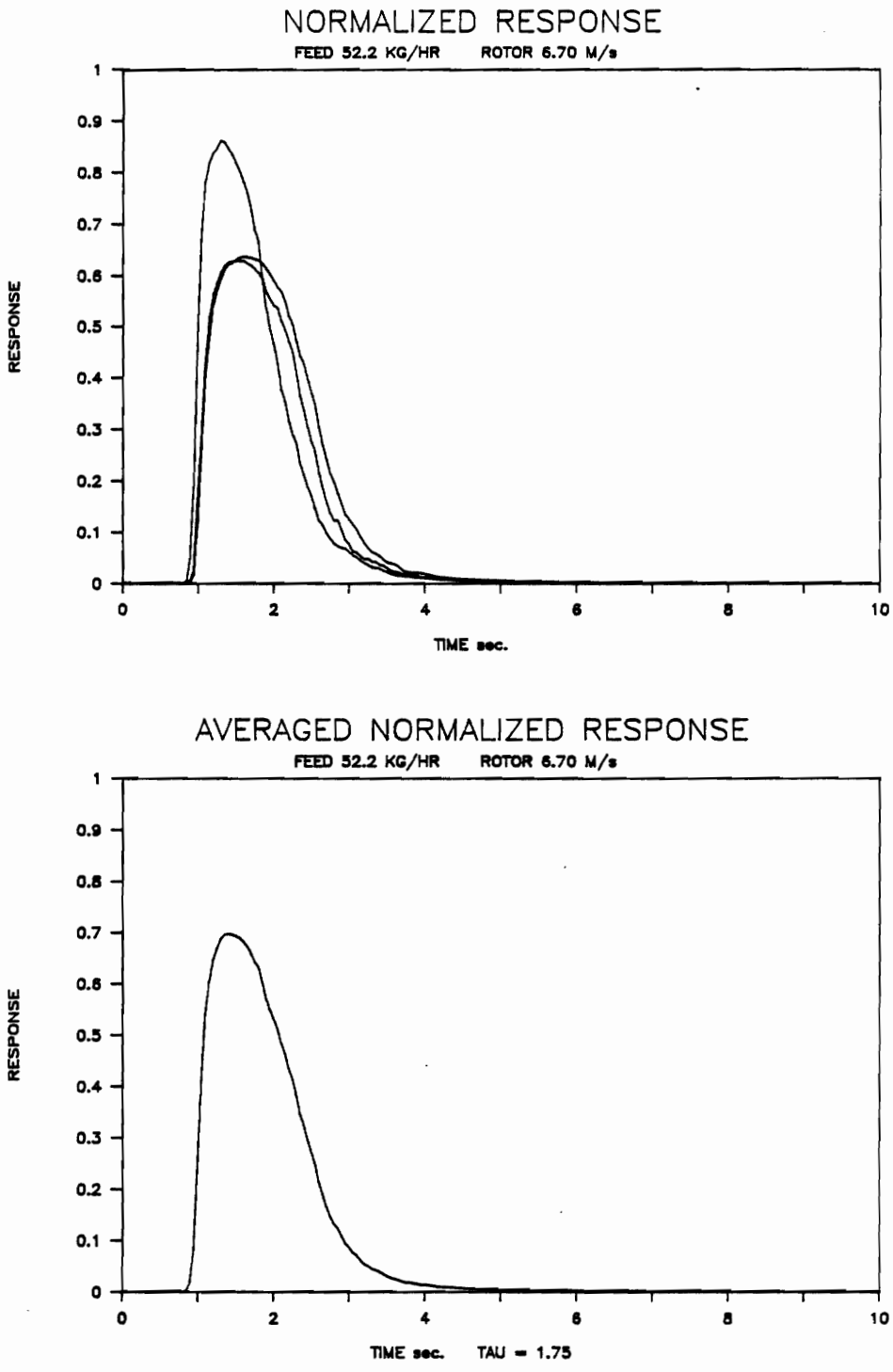
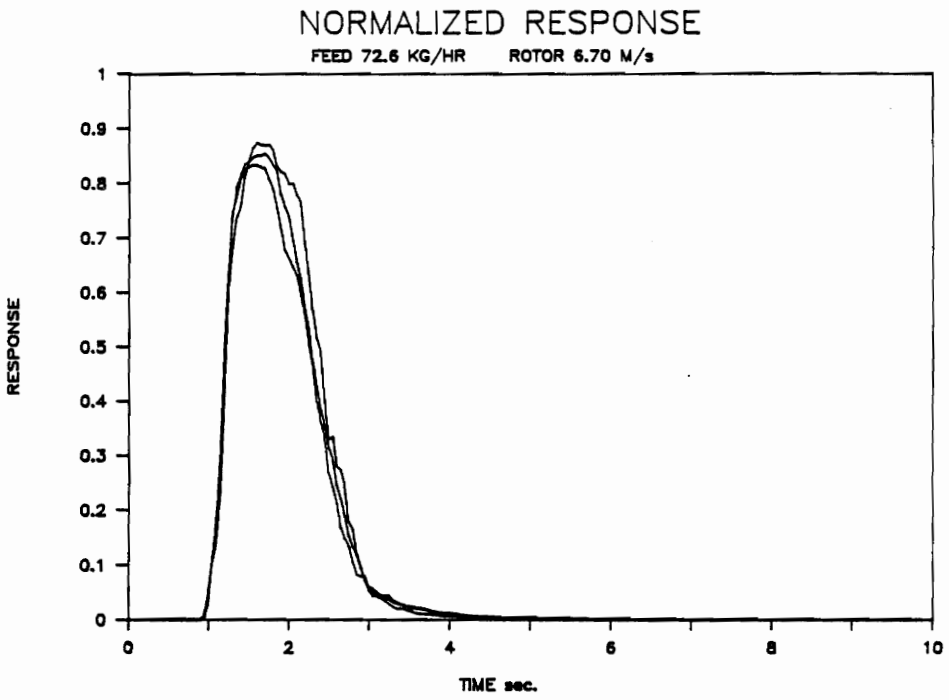


Figure A.1.9. Individual and averaged residence time distributions. TAU represents the mean residence time.



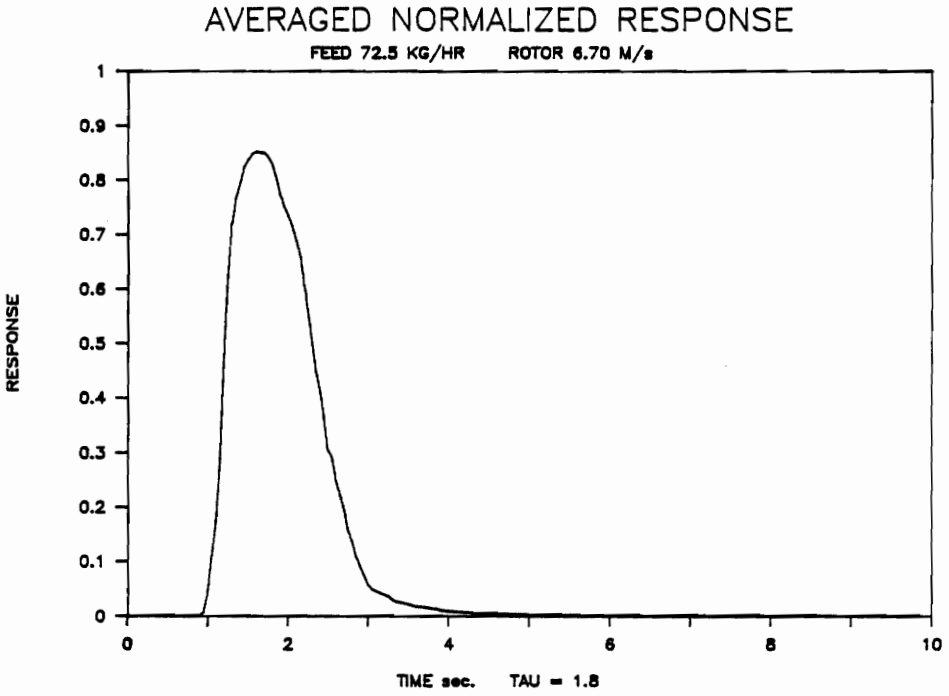


Figure A.1.10. Individual and averaged residence time distributions. TAU represents the mean residence time.

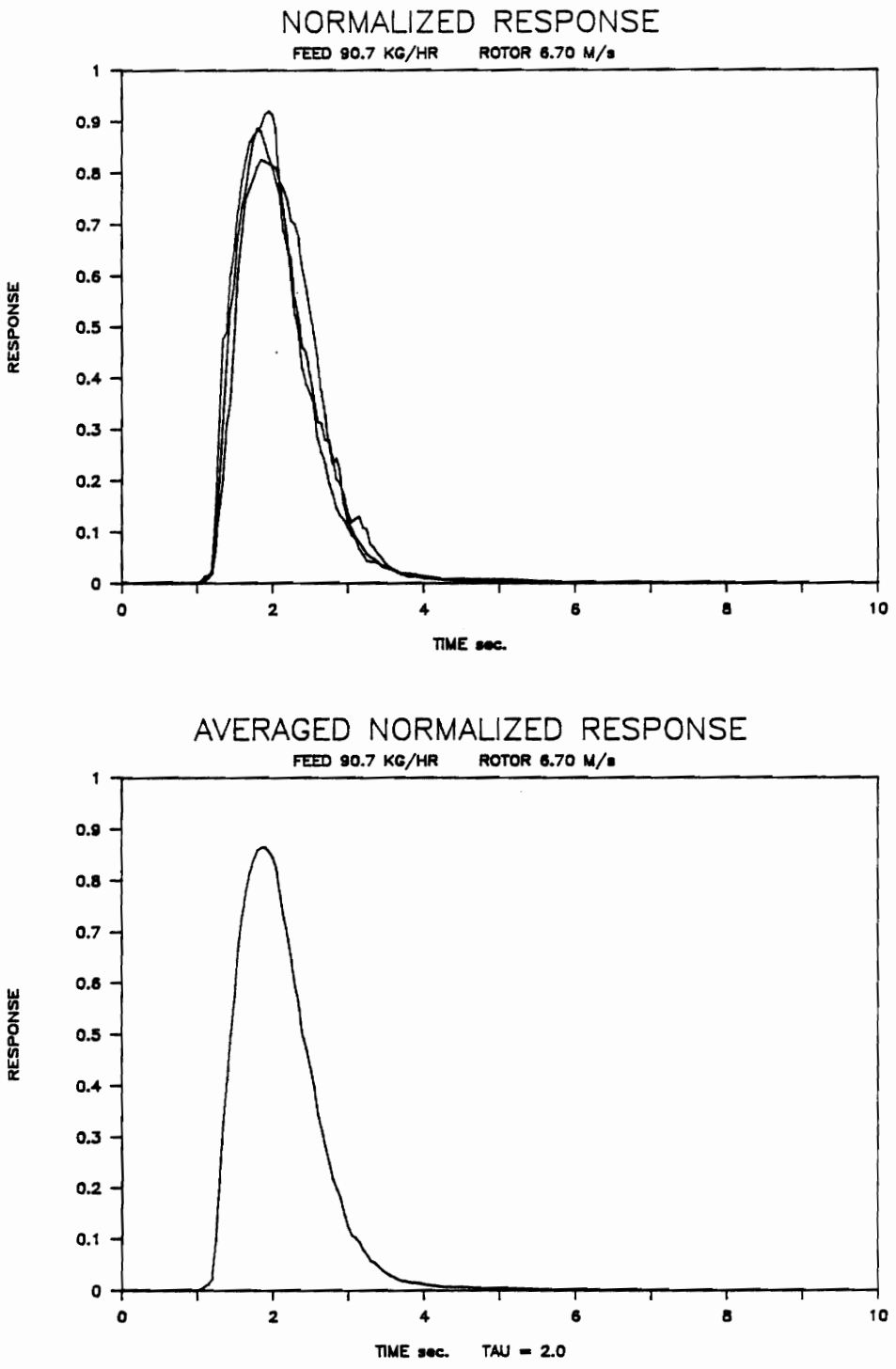


Figure A.1.11. Individual and averaged residence time distributions. TAU represents the mean residence time.

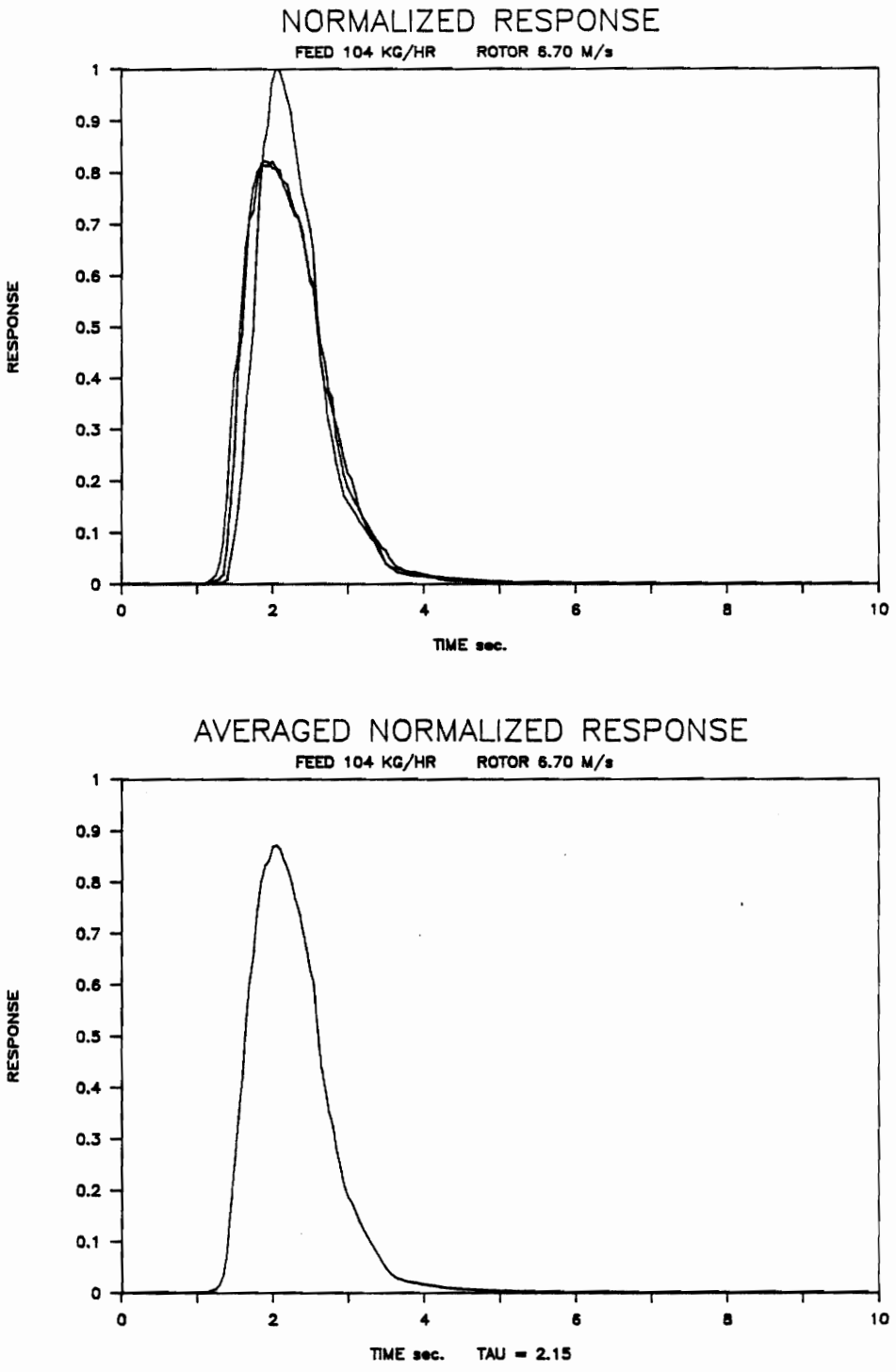


Figure A.1.12. Individual and averaged residence time distributions. TAU represents the mean residence time.

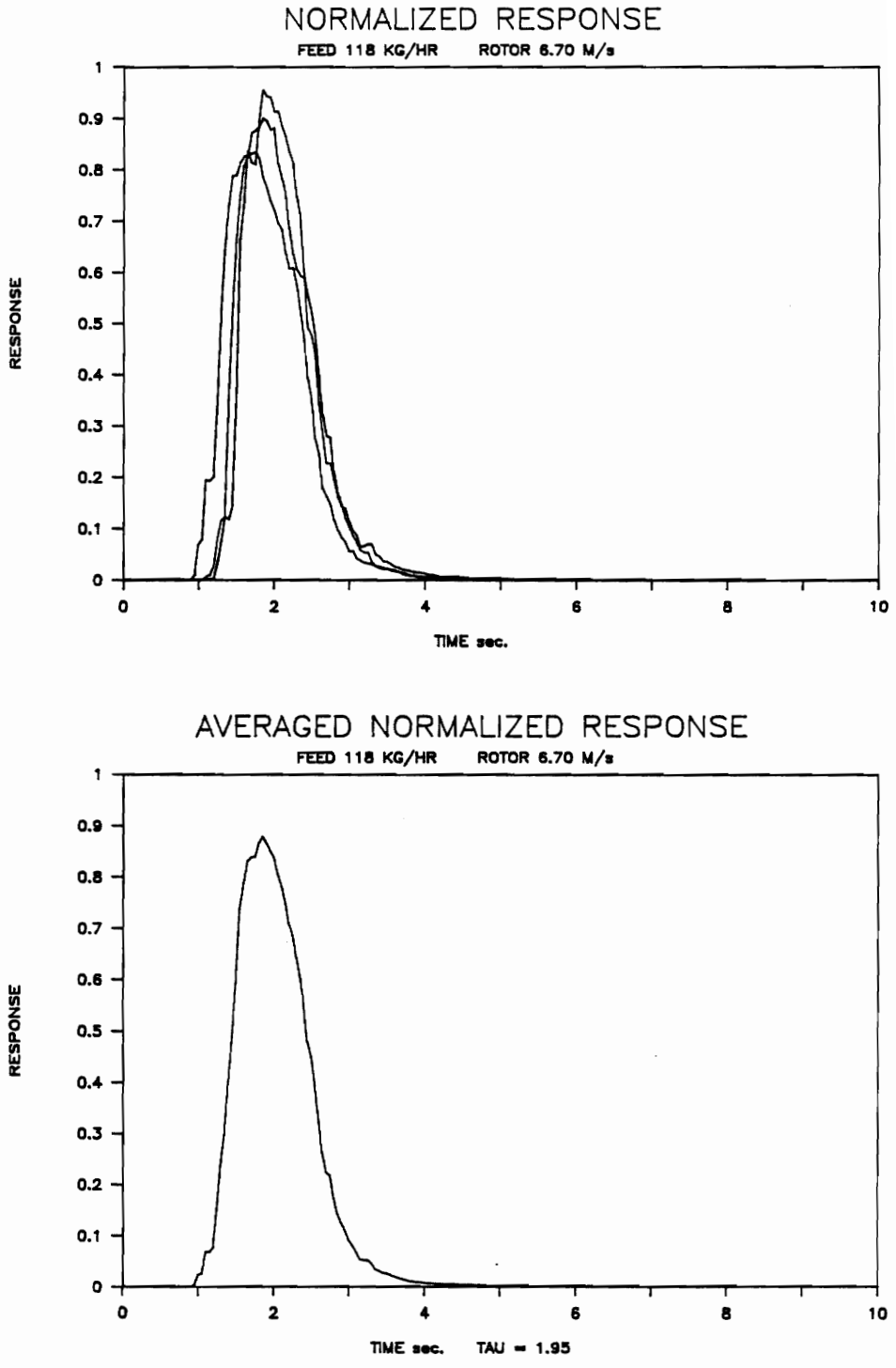


Figure A.1.13. Individual and averaged residence time distributions. TAU represents the mean residence time.

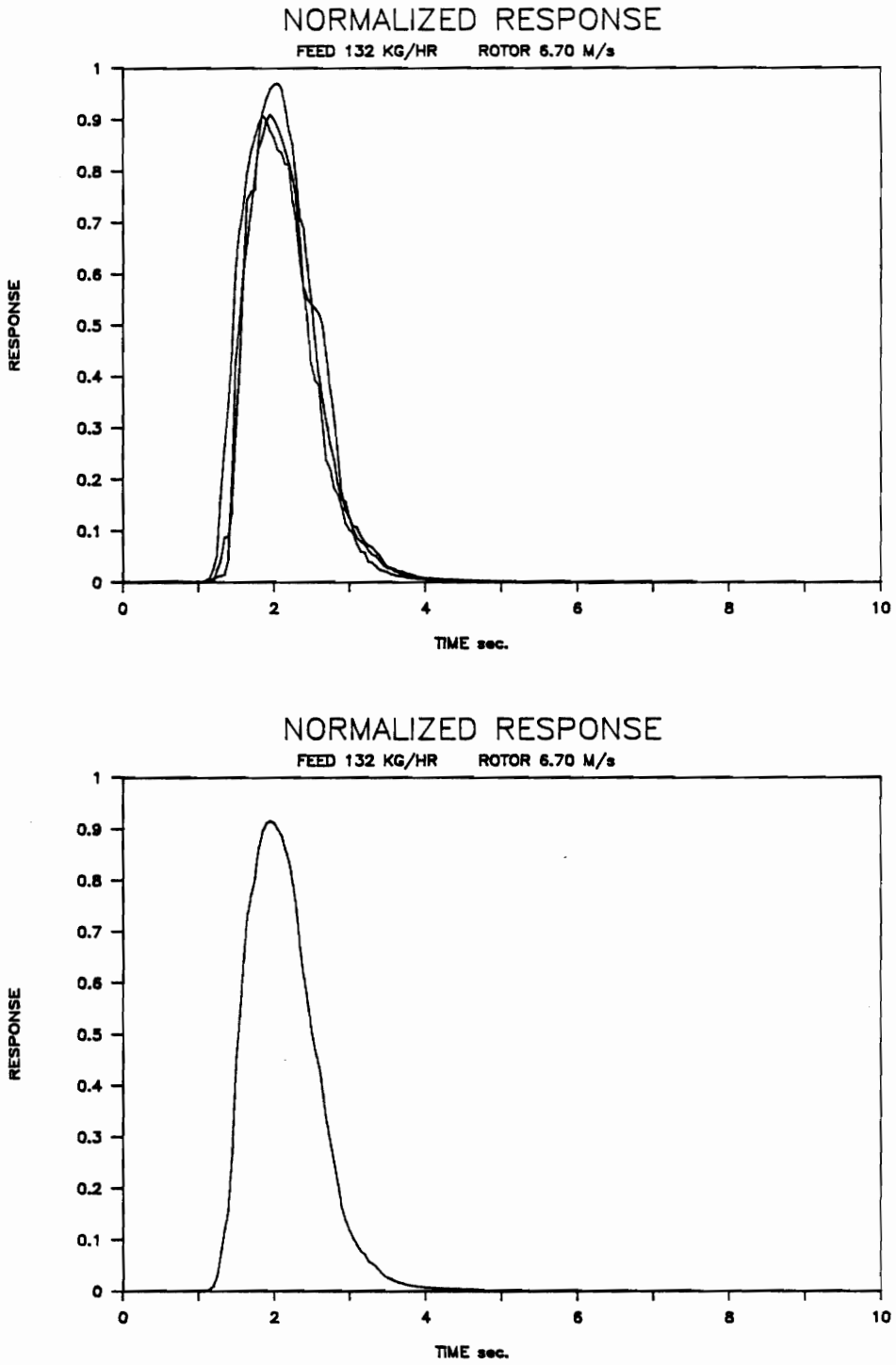


Figure A.1.14. Individual and averaged residence time distributions. TAU represents the mean residence time.

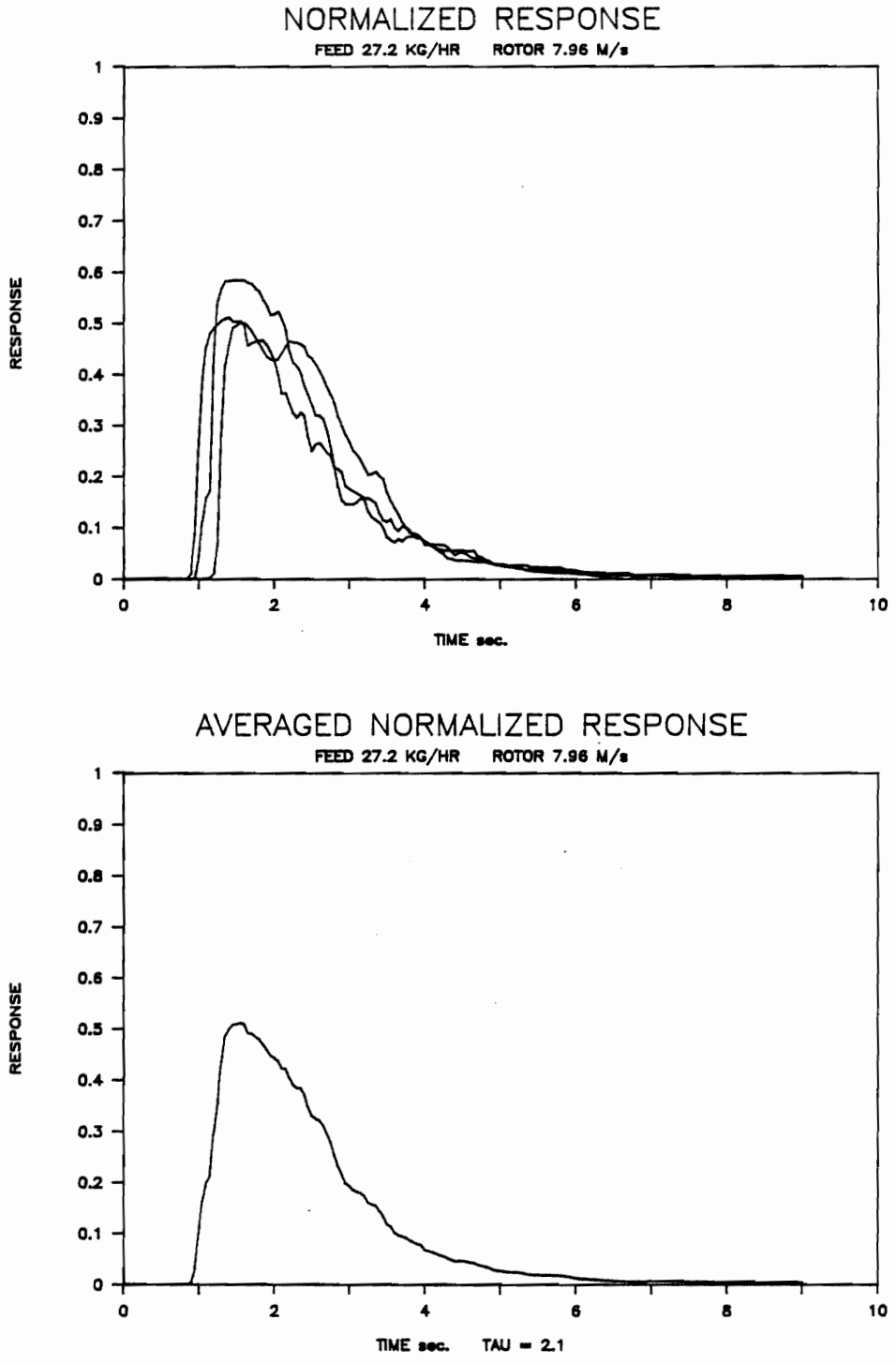


Figure A.1.15. Individual and averaged residence time distributions. TAU represents the mean residence time.

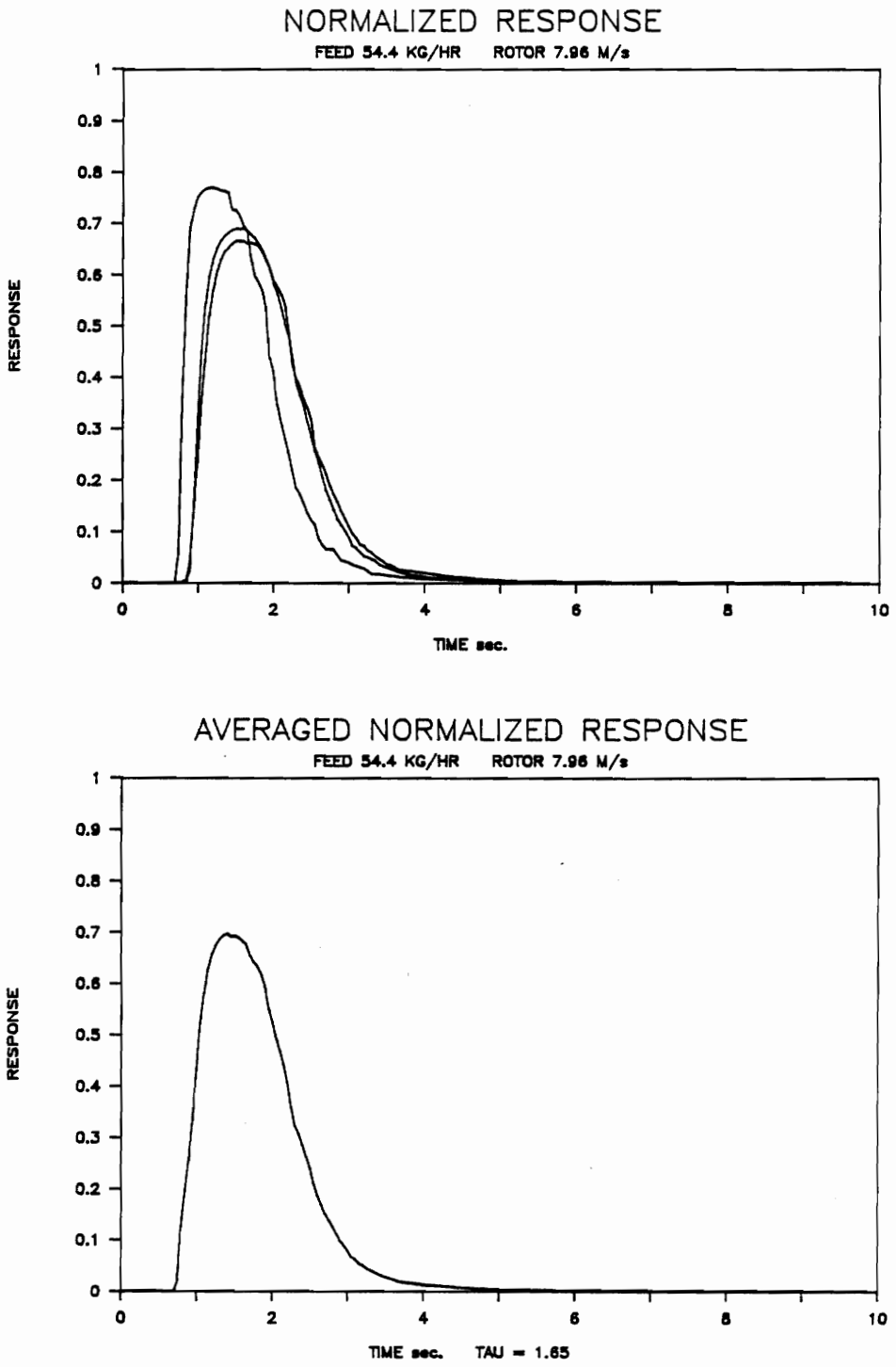
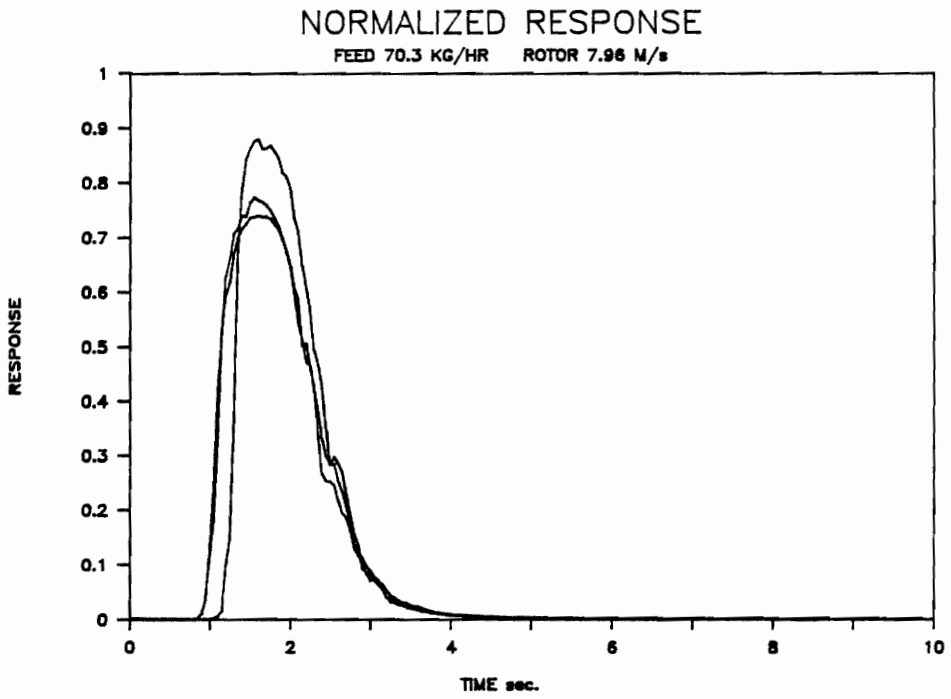


Figure A.1.16. Individual and averaged residence time distributions. TAU represents the mean residence time.



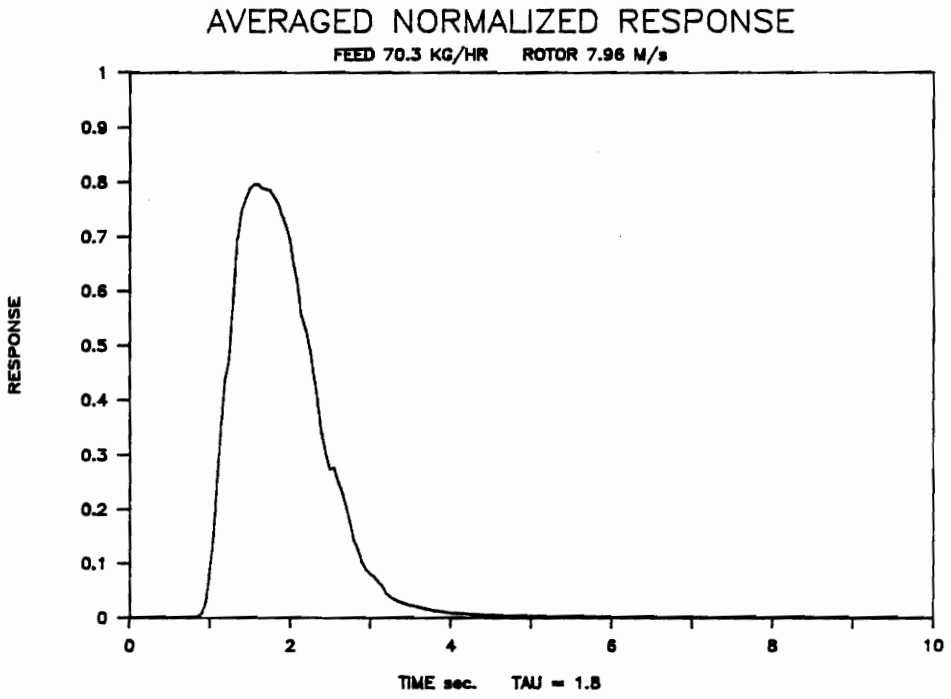


Figure A.1.17. Individual and averaged residence time distributions. TAU represents the mean residence time.

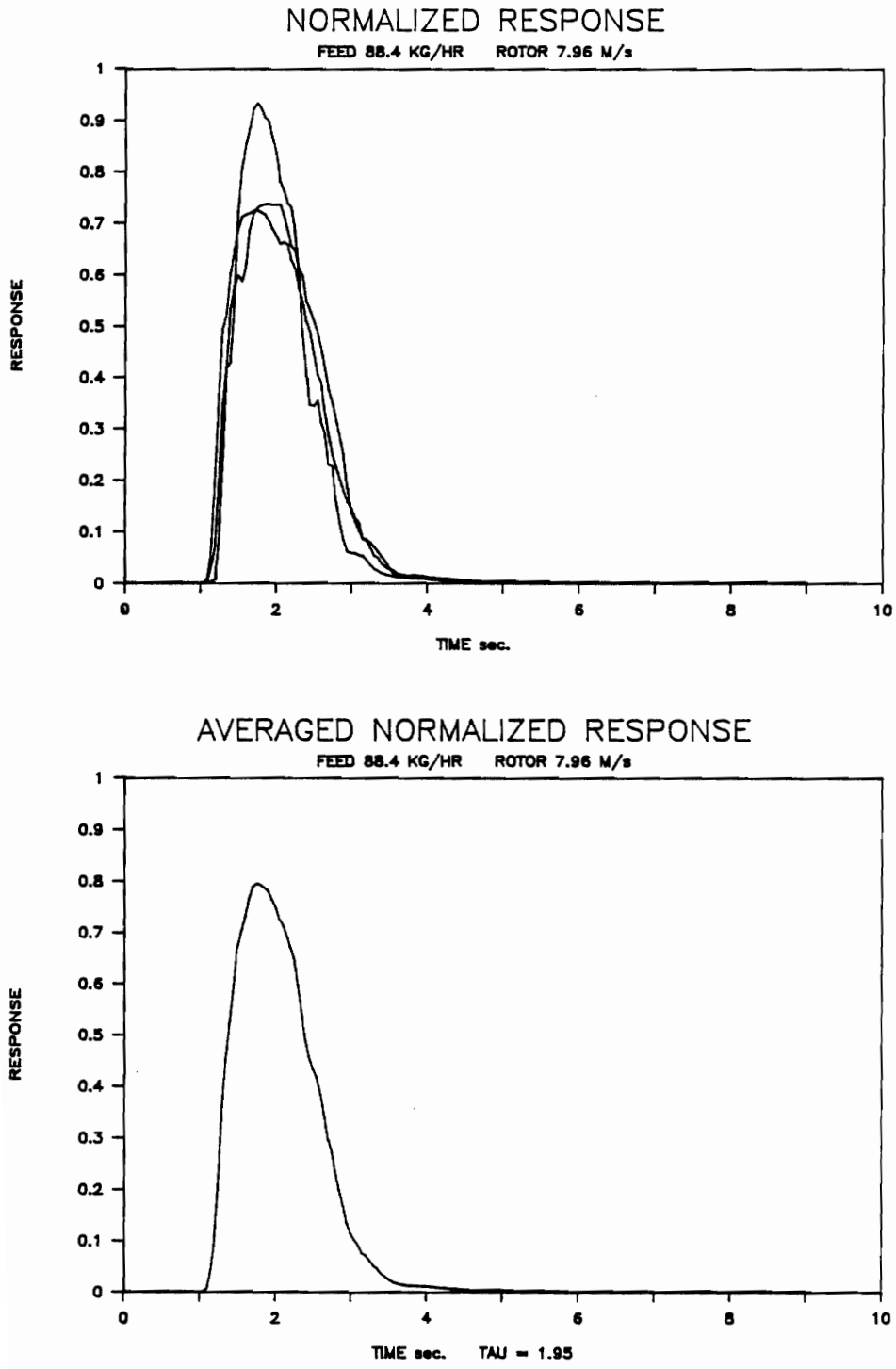


Figure A.1.18. Individual and averaged residence time distributions. TAU represents the mean residence time.

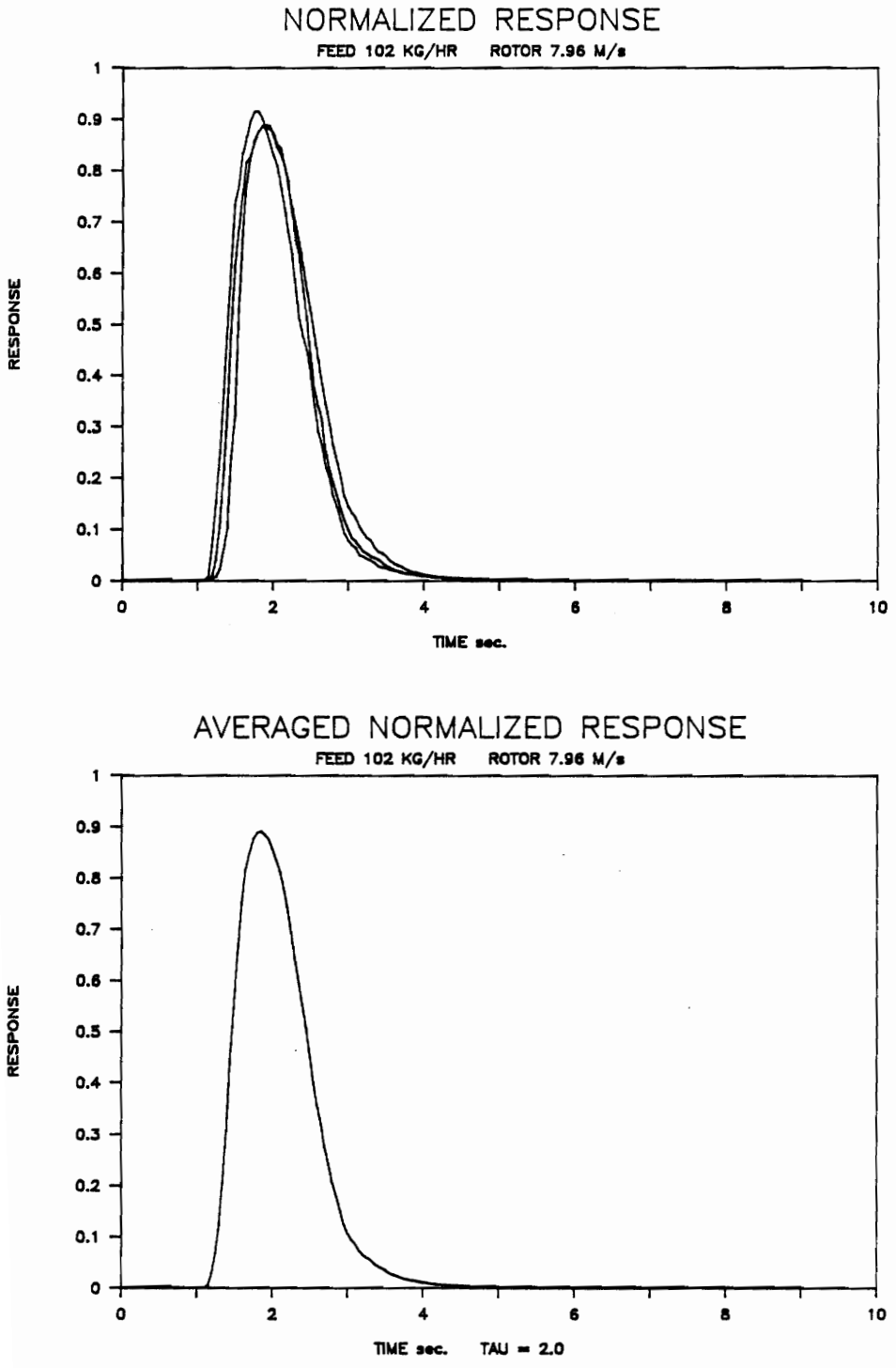


Figure A.1.19. Individual and averaged residence time distributions. TAU represents the mean residence time.

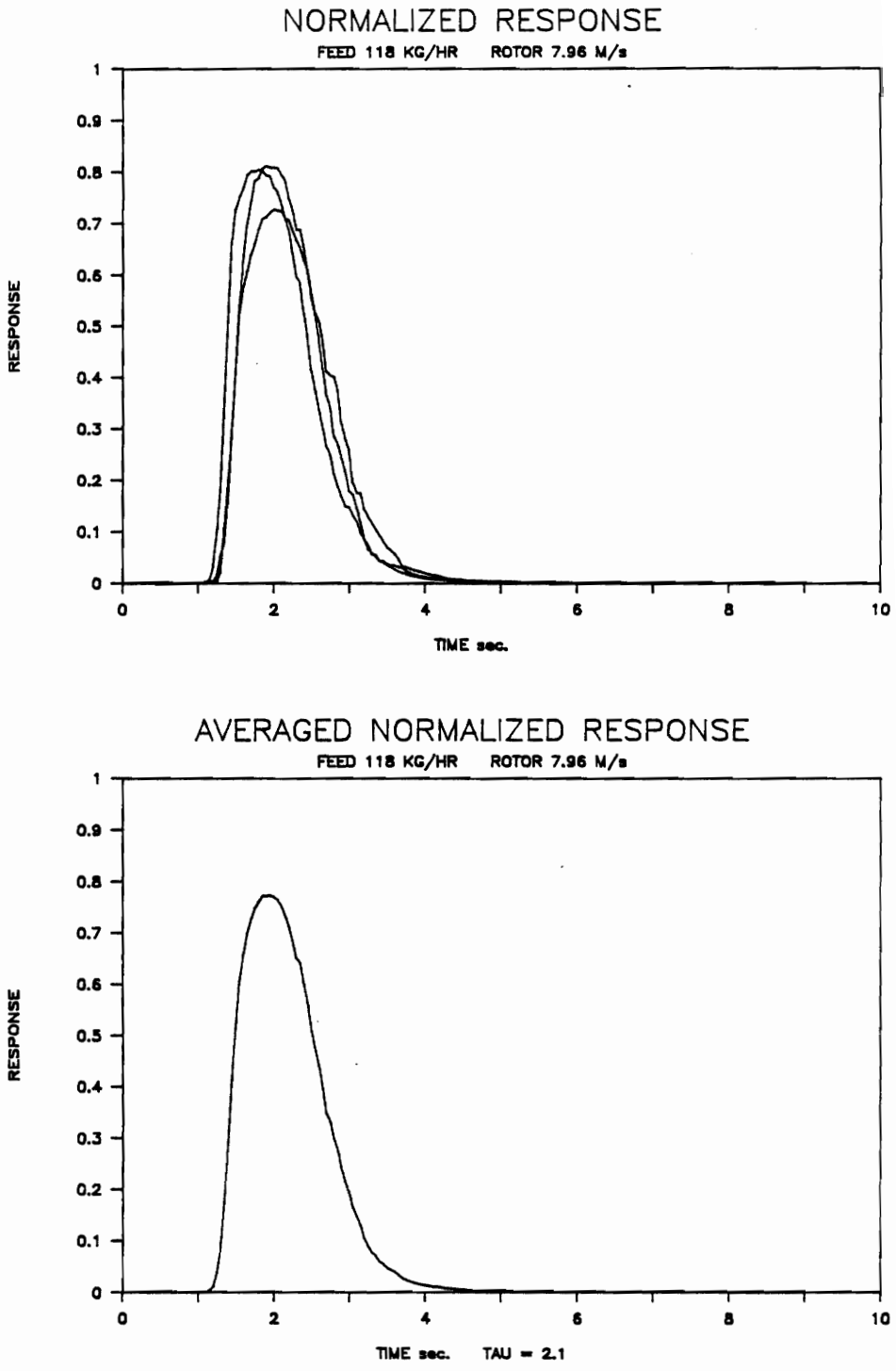


Figure A.1.20. Individual and averaged residence time distributions. TAU represents the mean residence time.

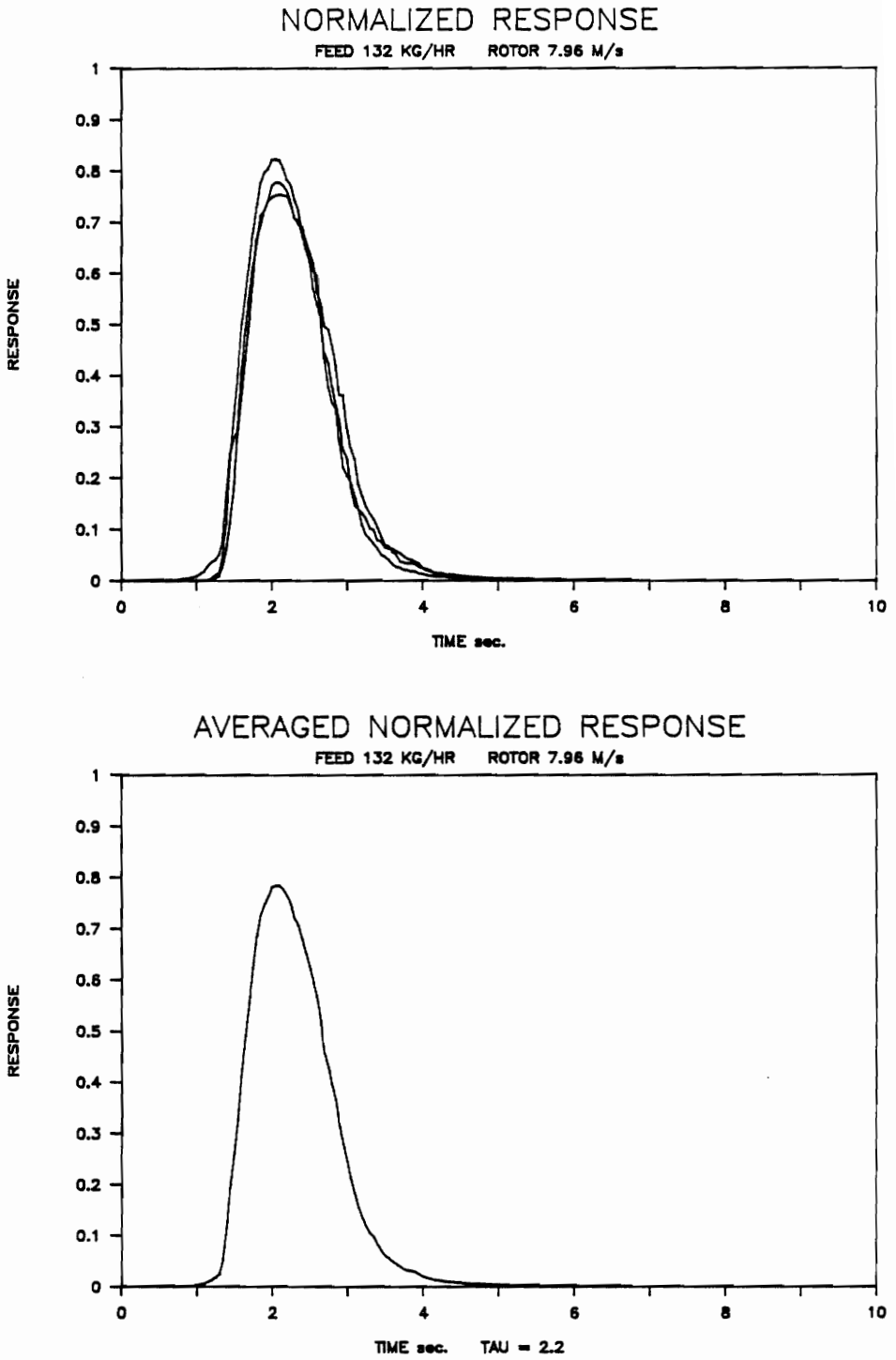


Figure A.1.21. Individual and averaged residence time distributions. TAU represents the mean residence time.

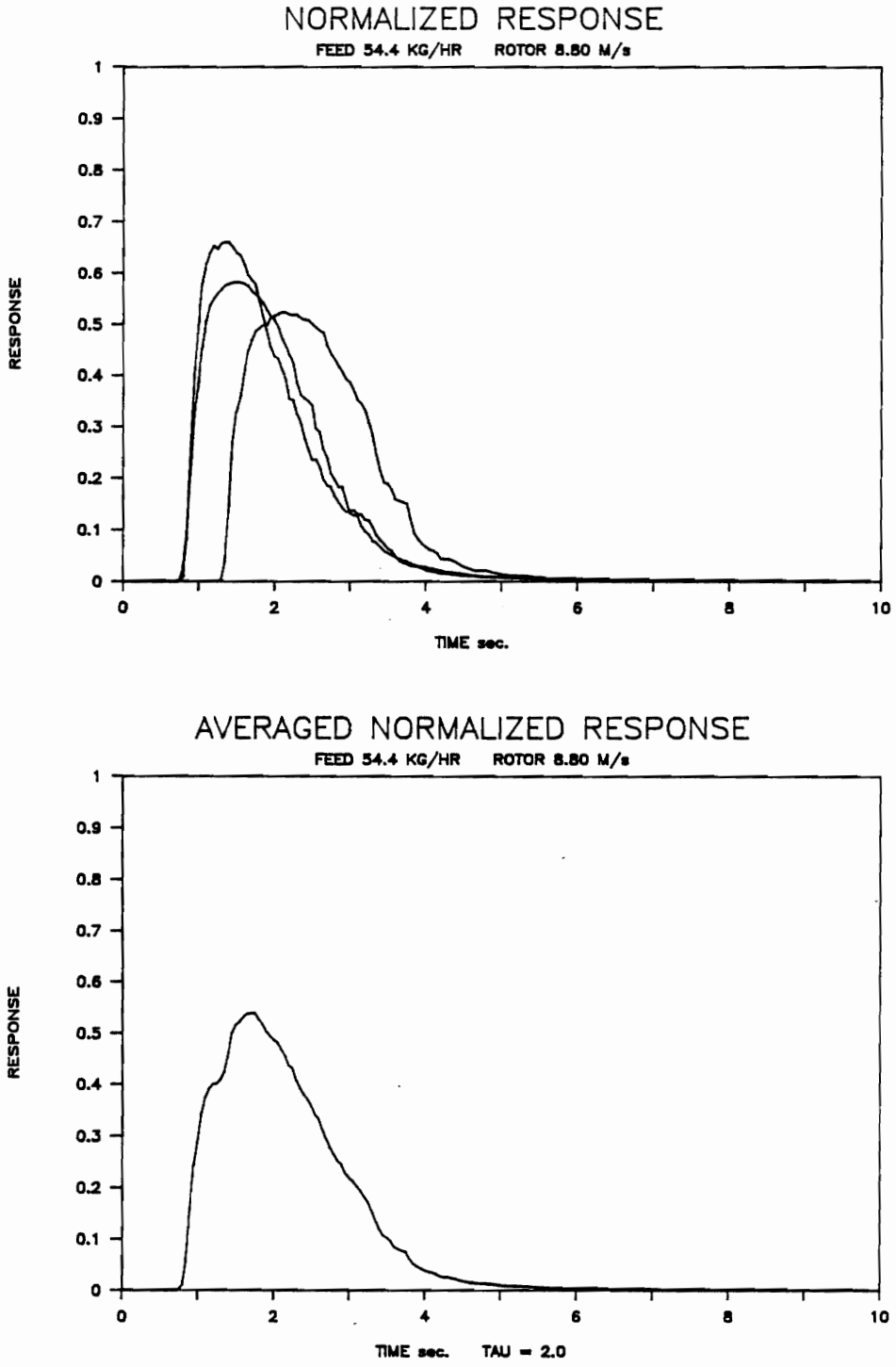


Figure A.1.22. Individual and averaged residence time distributions. TAU represents the mean residence time.

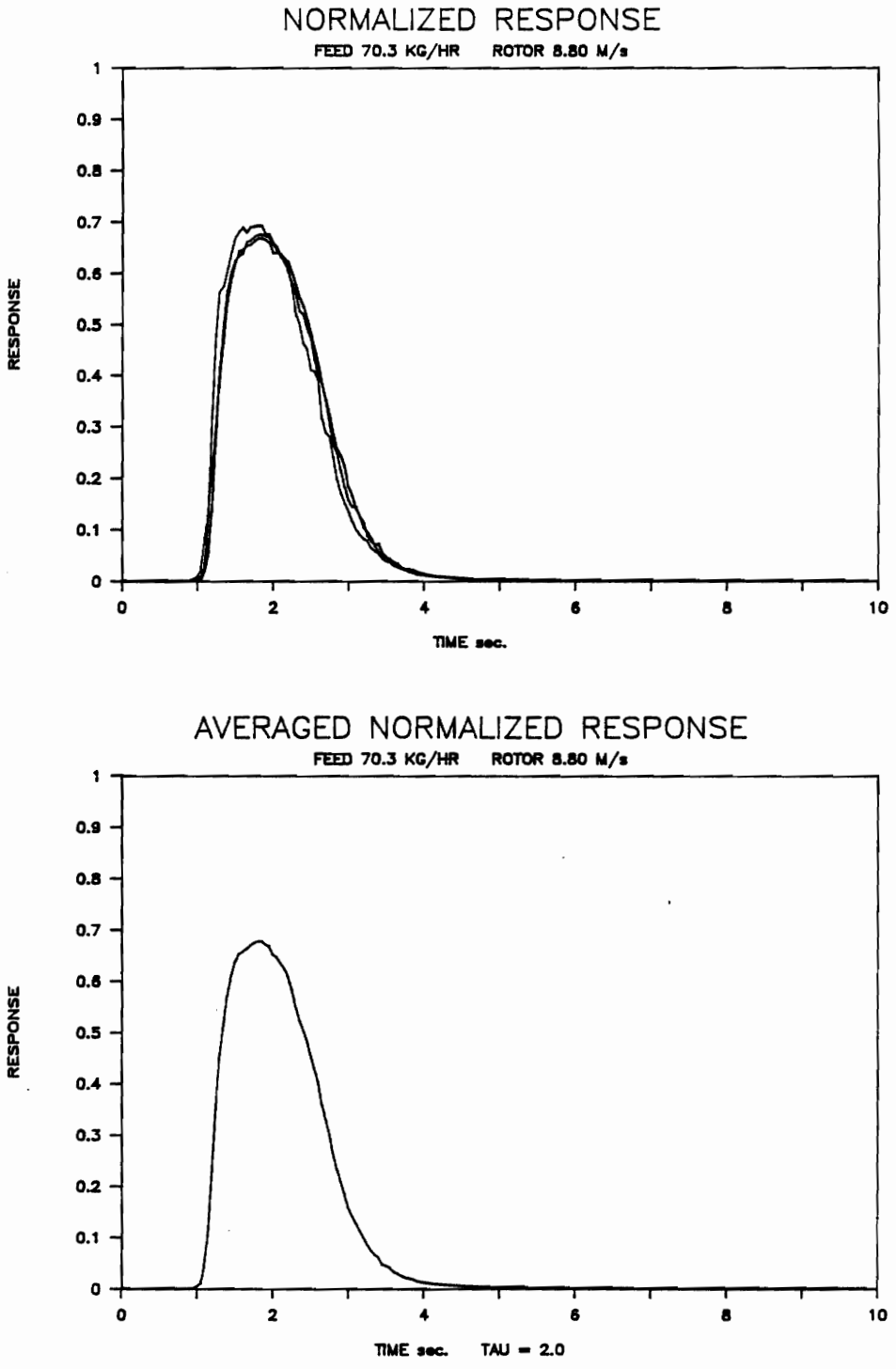


Figure A.1.23. Individual and averaged residence time distributions. TAU represents the mean residence time.

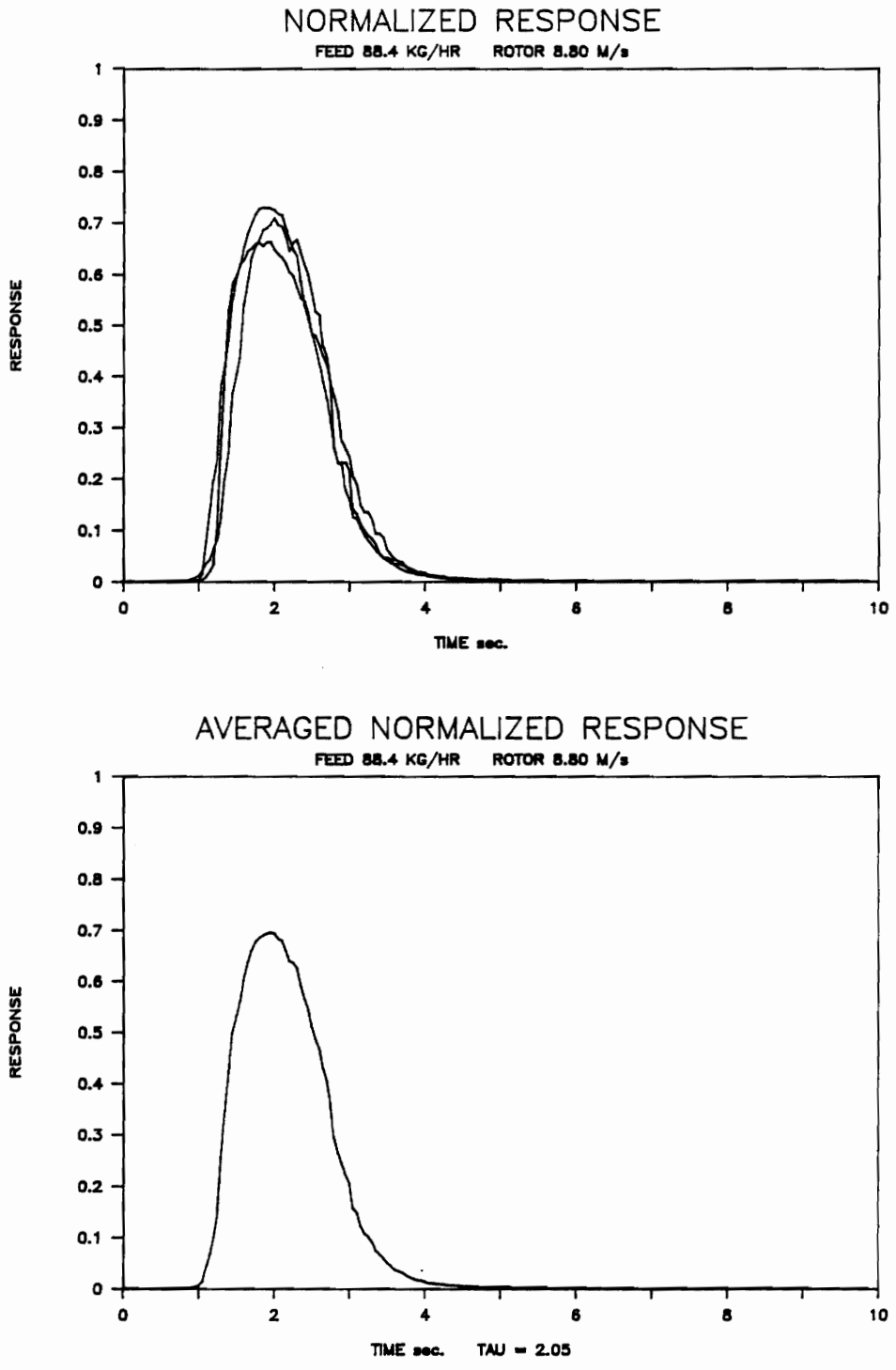


Figure A.1.24. Individual and averaged residence time distributions. TAU represents the mean residence time.

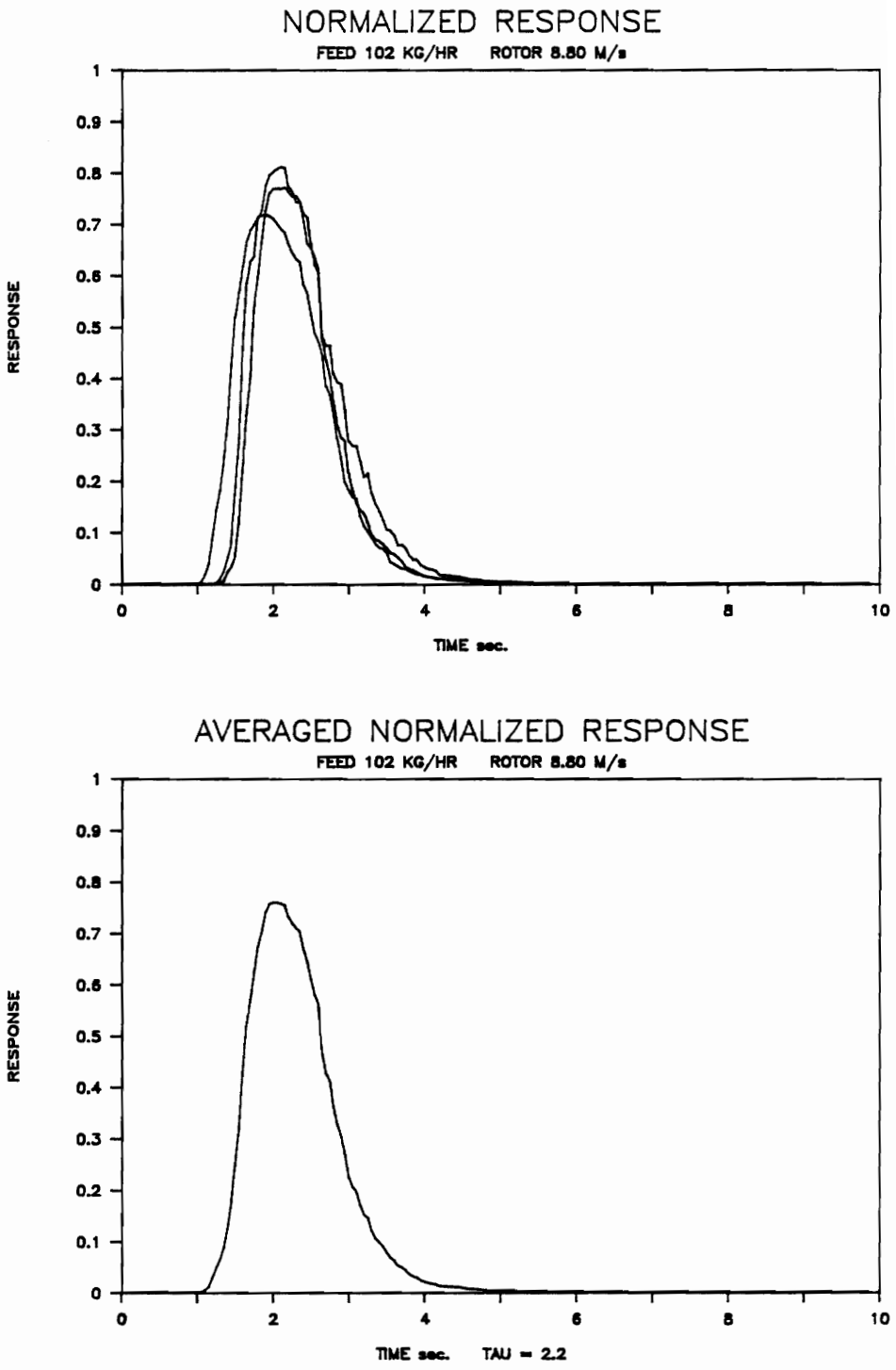


Figure A.1.25. Individual and averaged residence time distributions. TAU represents the mean residence time.

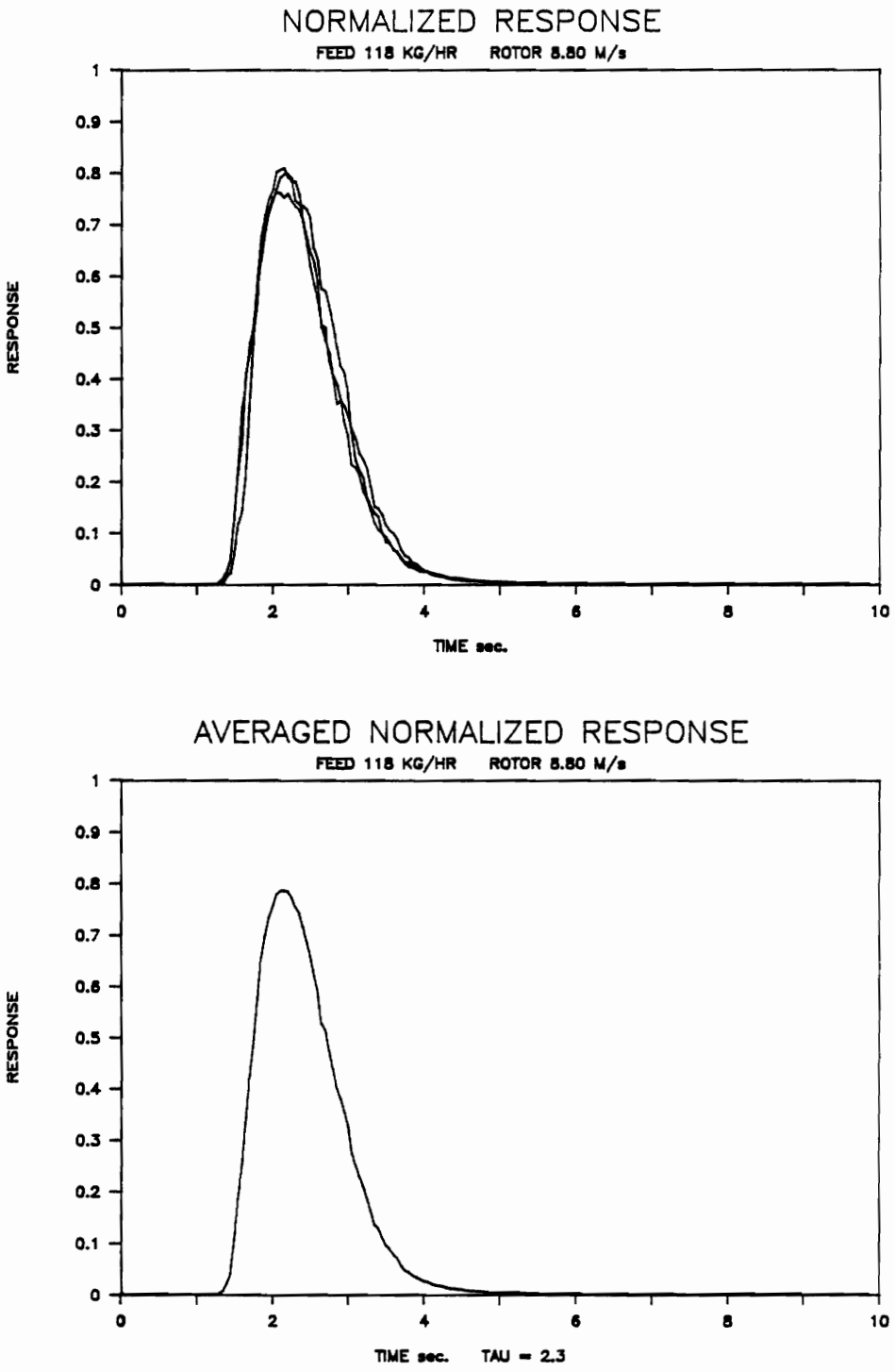


Figure A.1.26. Individual and averaged residence time distributions. TAU represents the mean residence time.

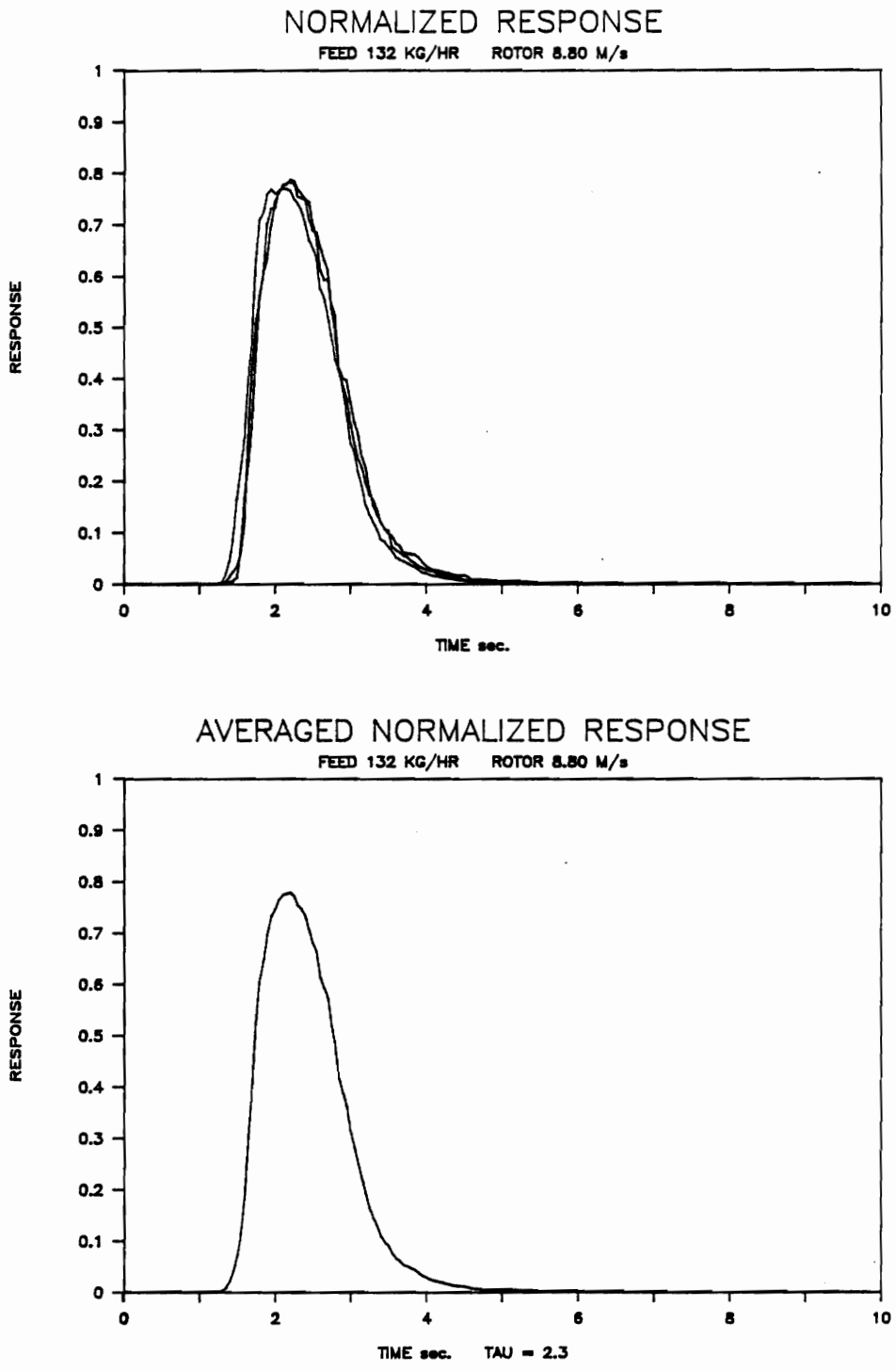


Figure A.1.27. Individual and averaged residence time distributions. TAU represents the mean residence time.

APPENDIX 2

MODEL CURVE-FIT PLOTS

Note: 'N' denotes the number of model stages.

'Dt' denotes the model dead time.

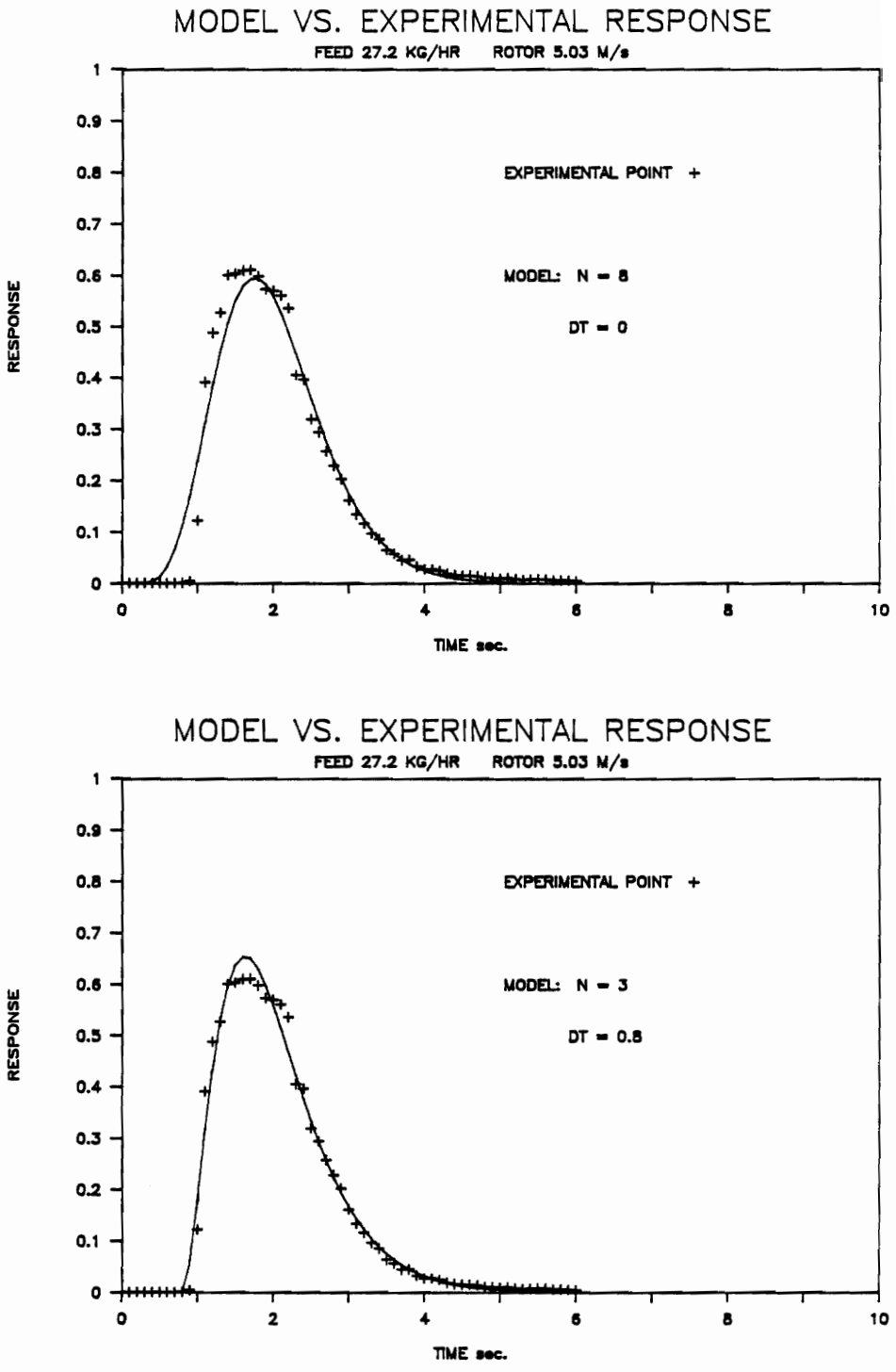


Figure A.2.1. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

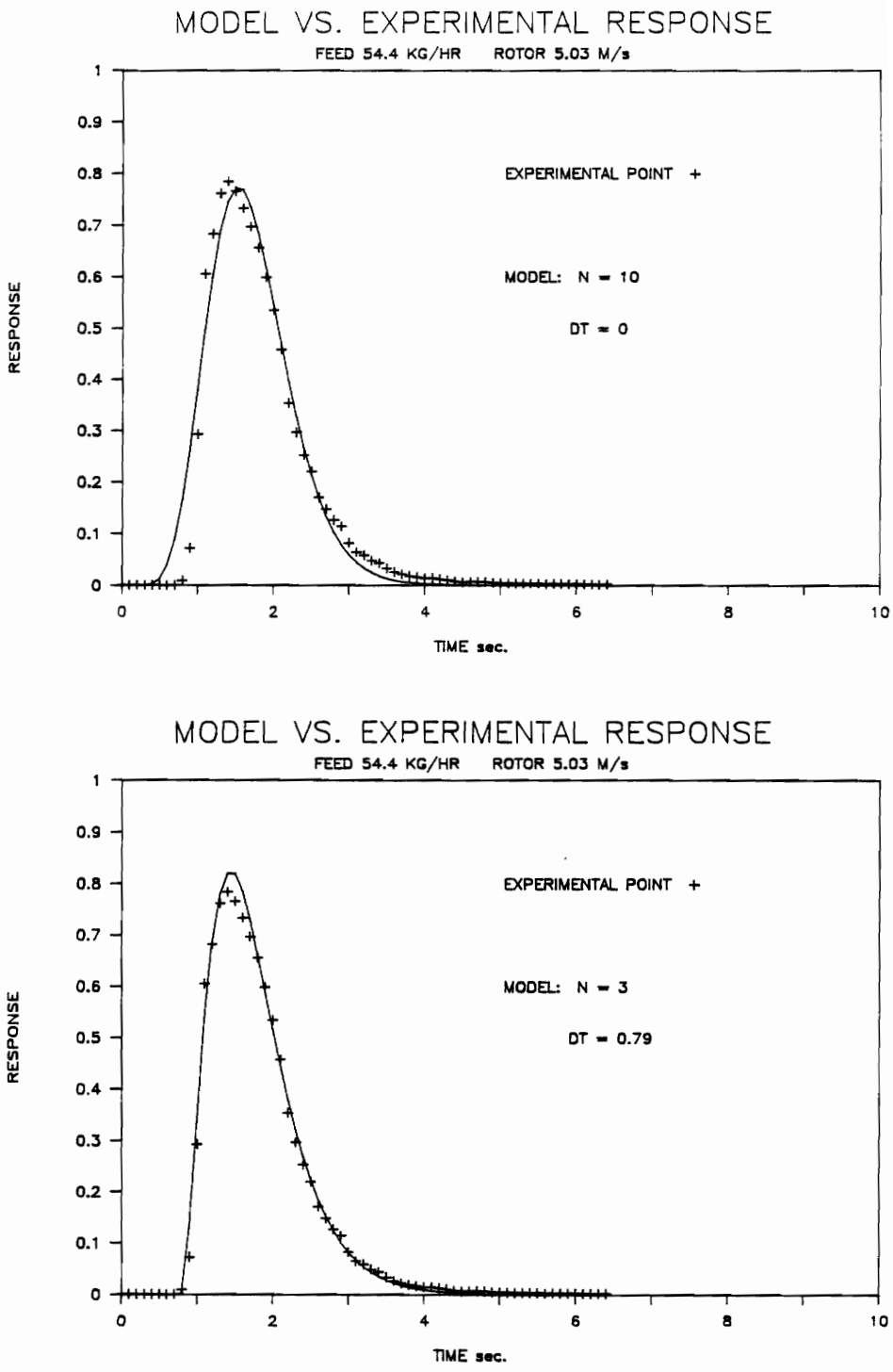
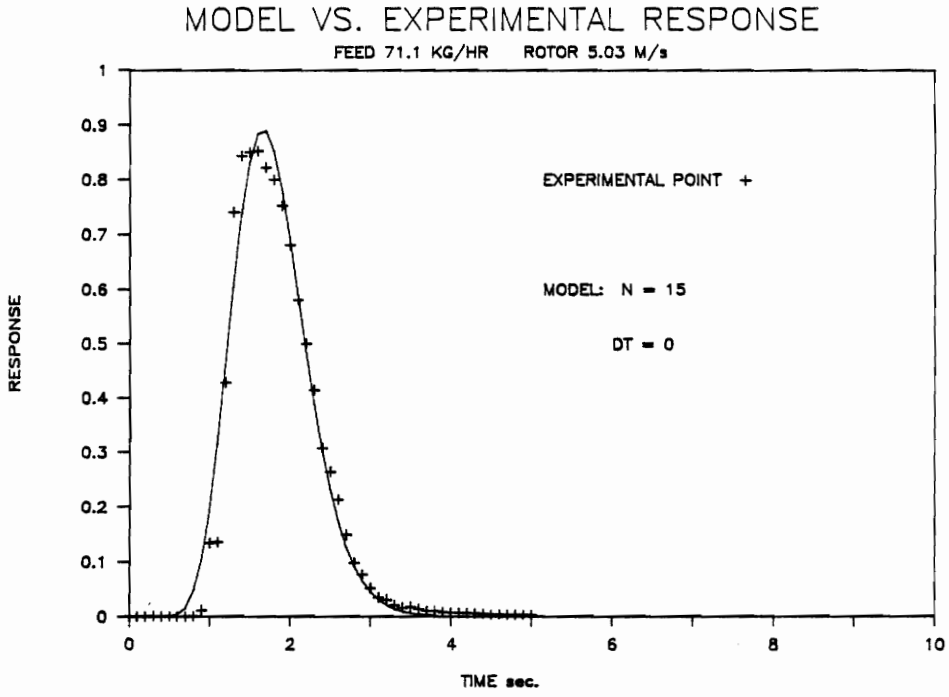


Figure A.2.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



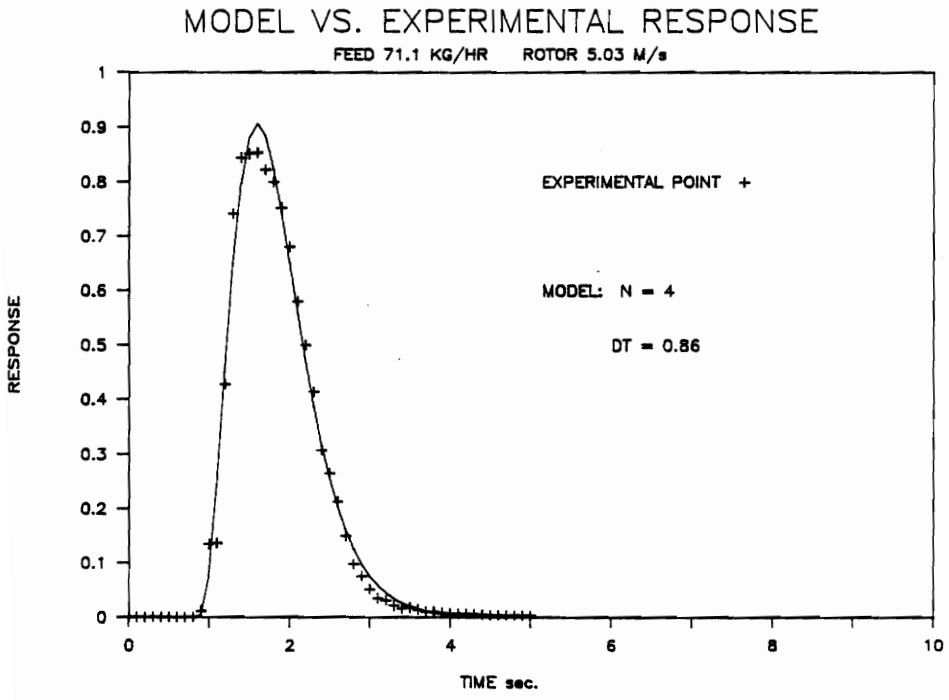


Figure A.2.1. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

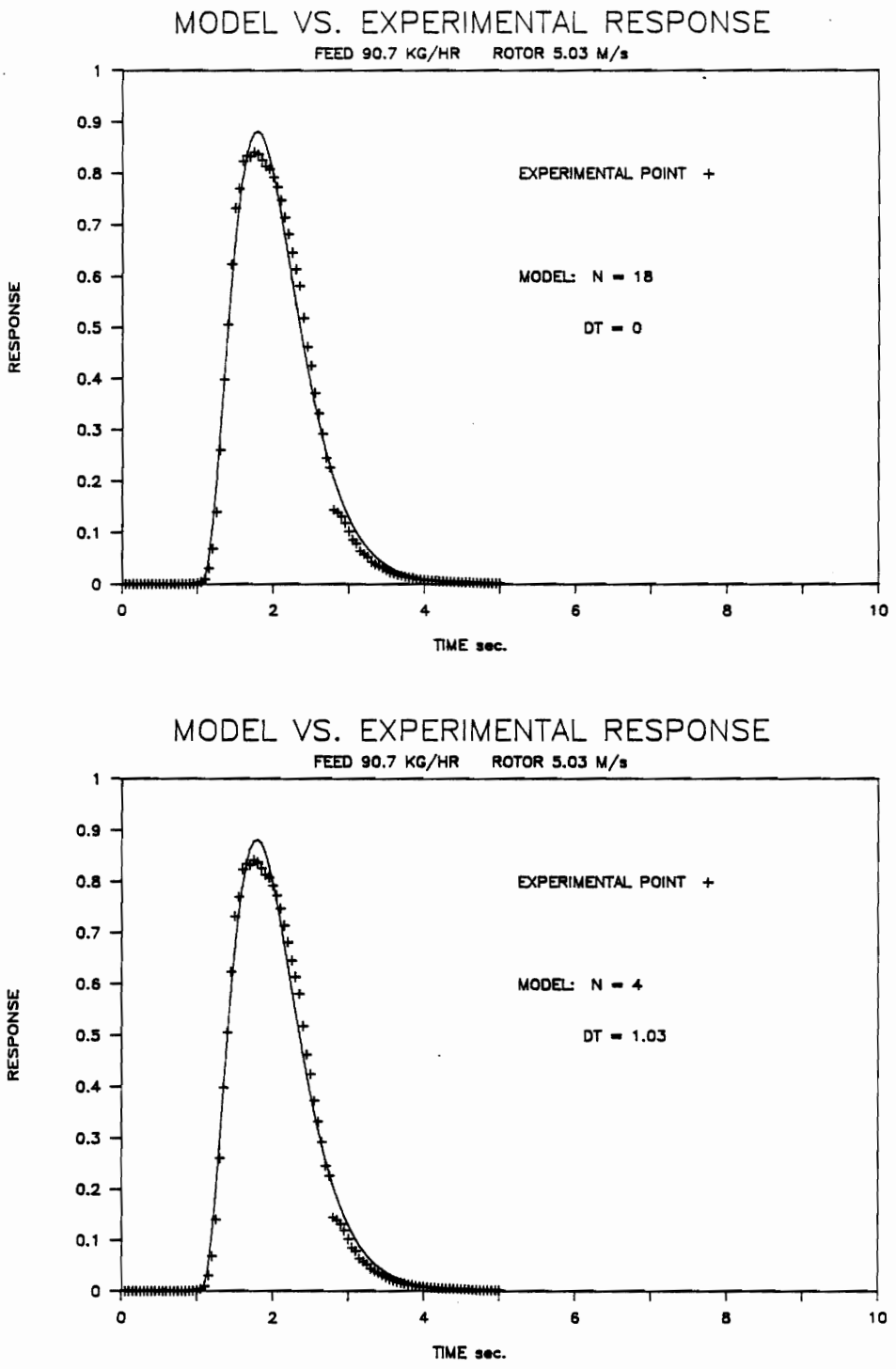


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

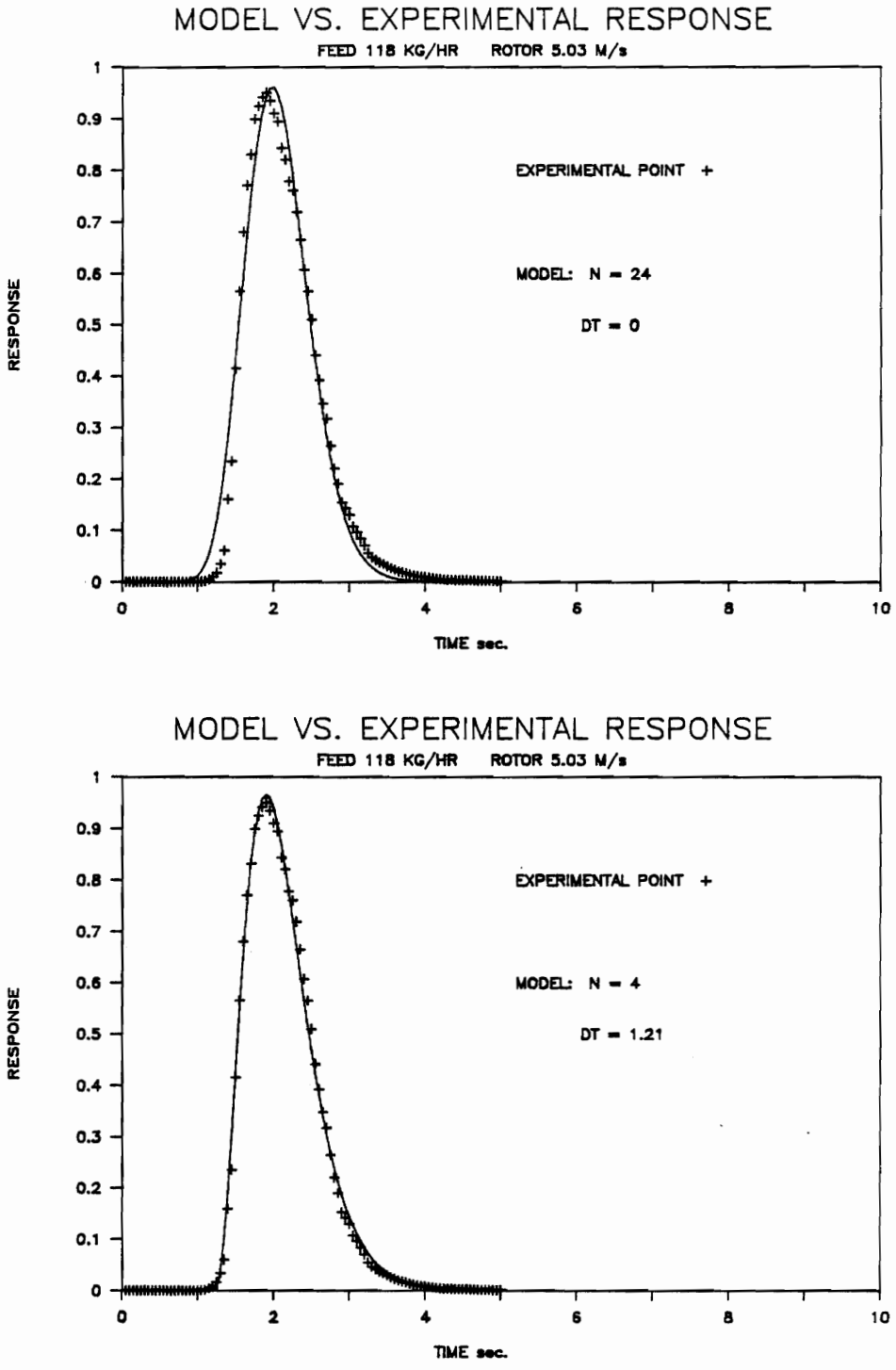
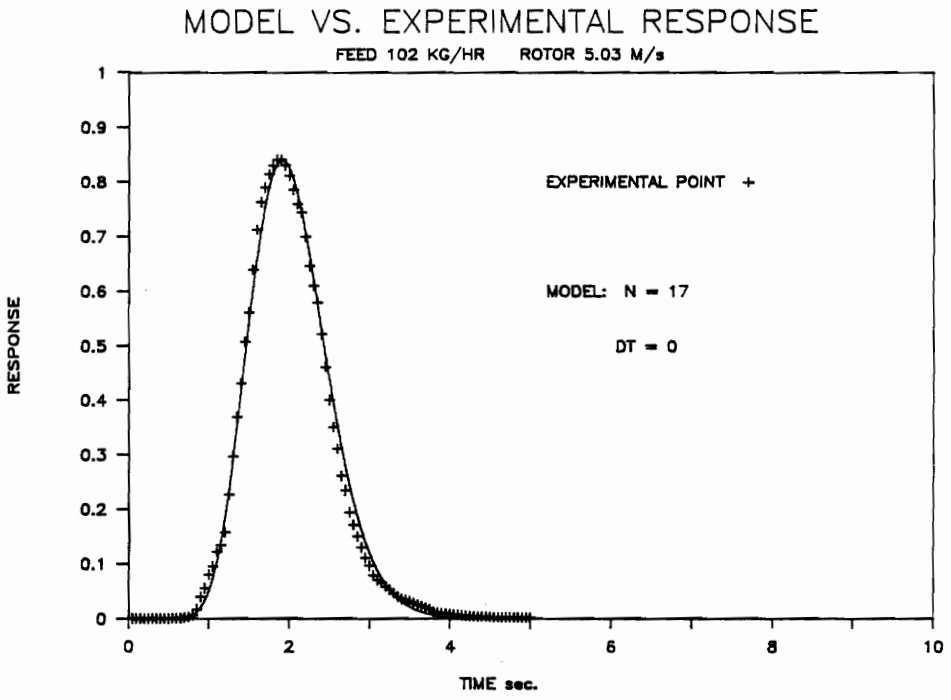


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



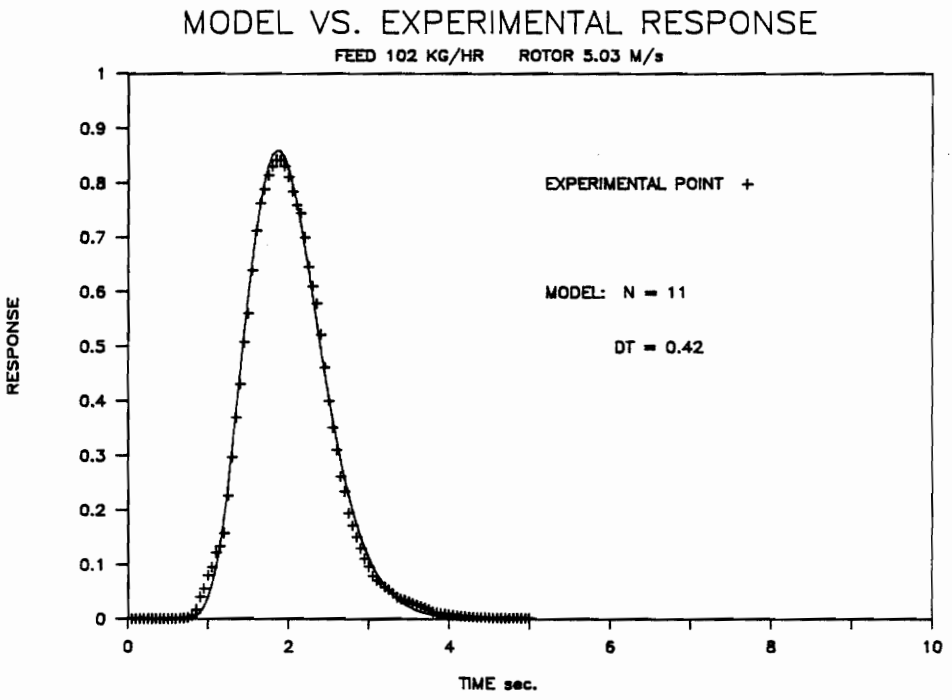


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

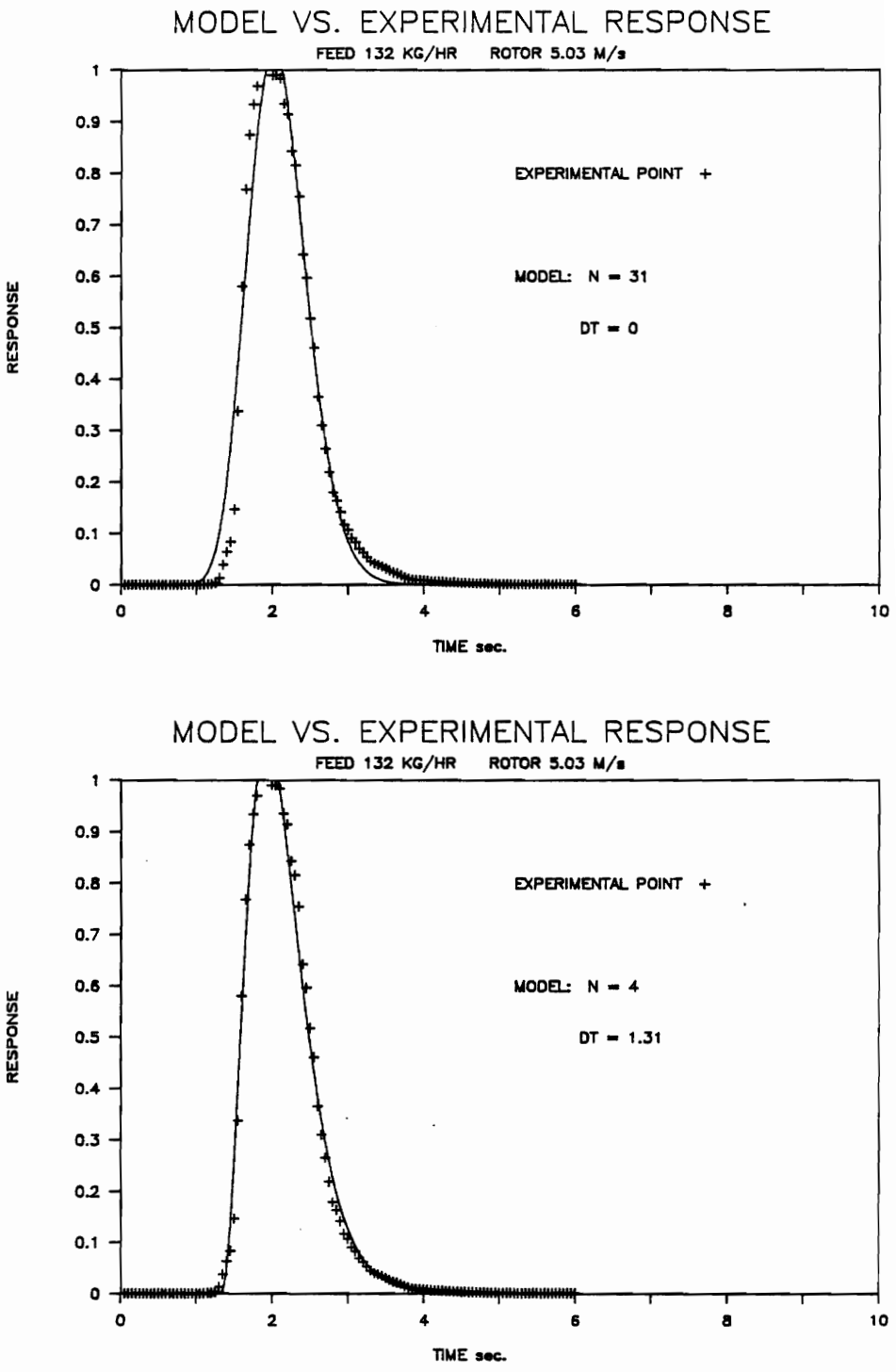


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

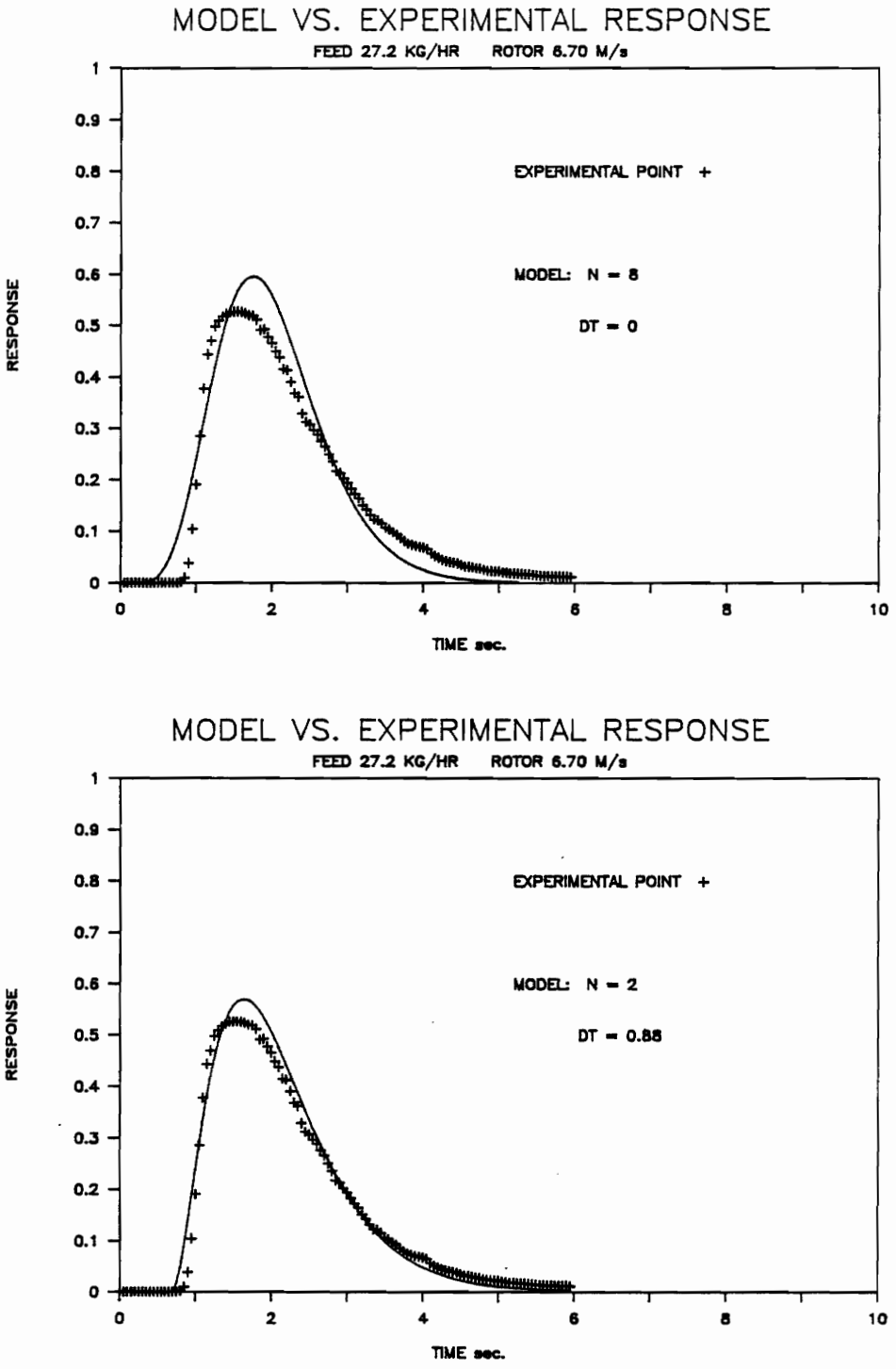


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

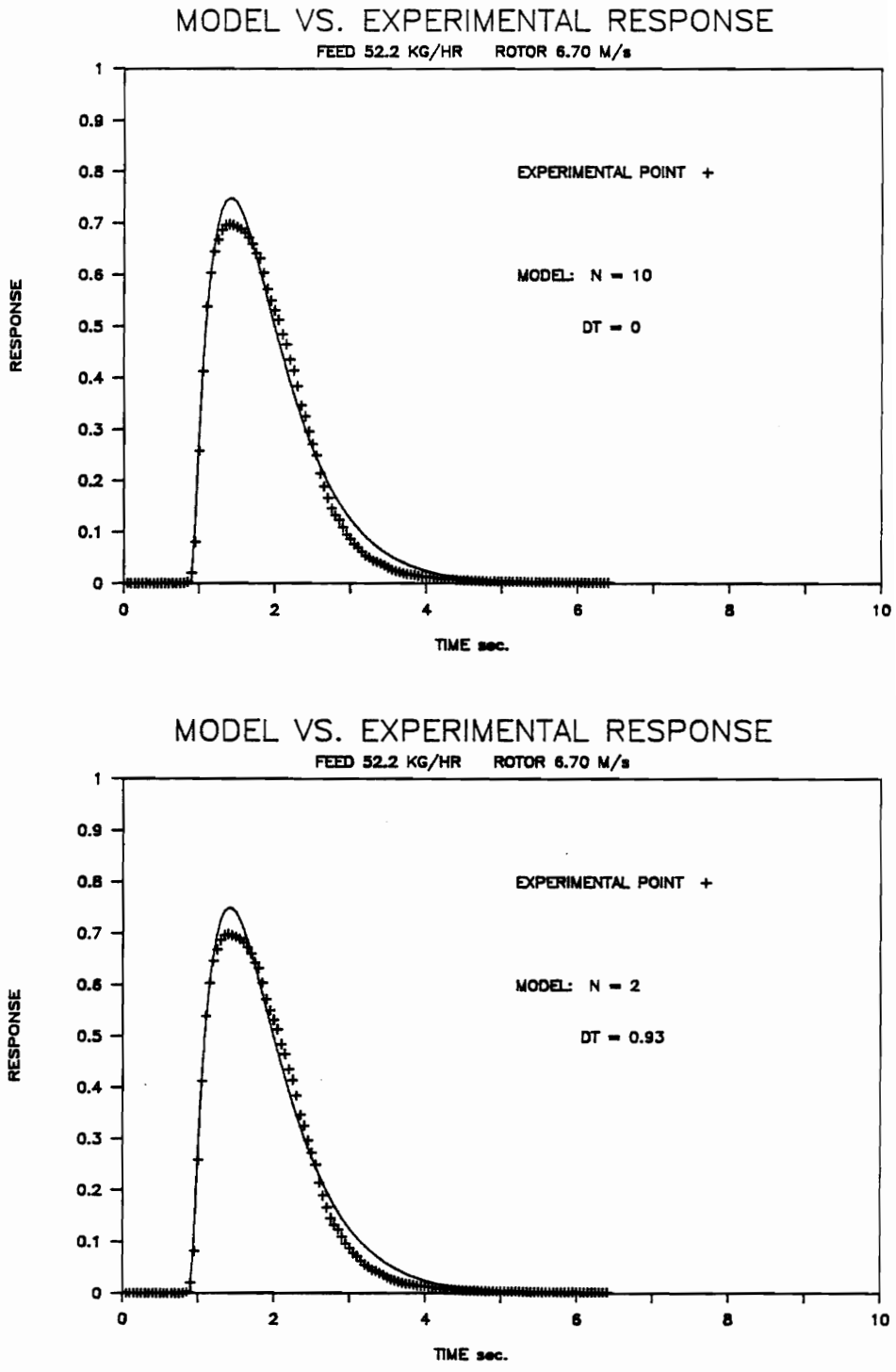


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

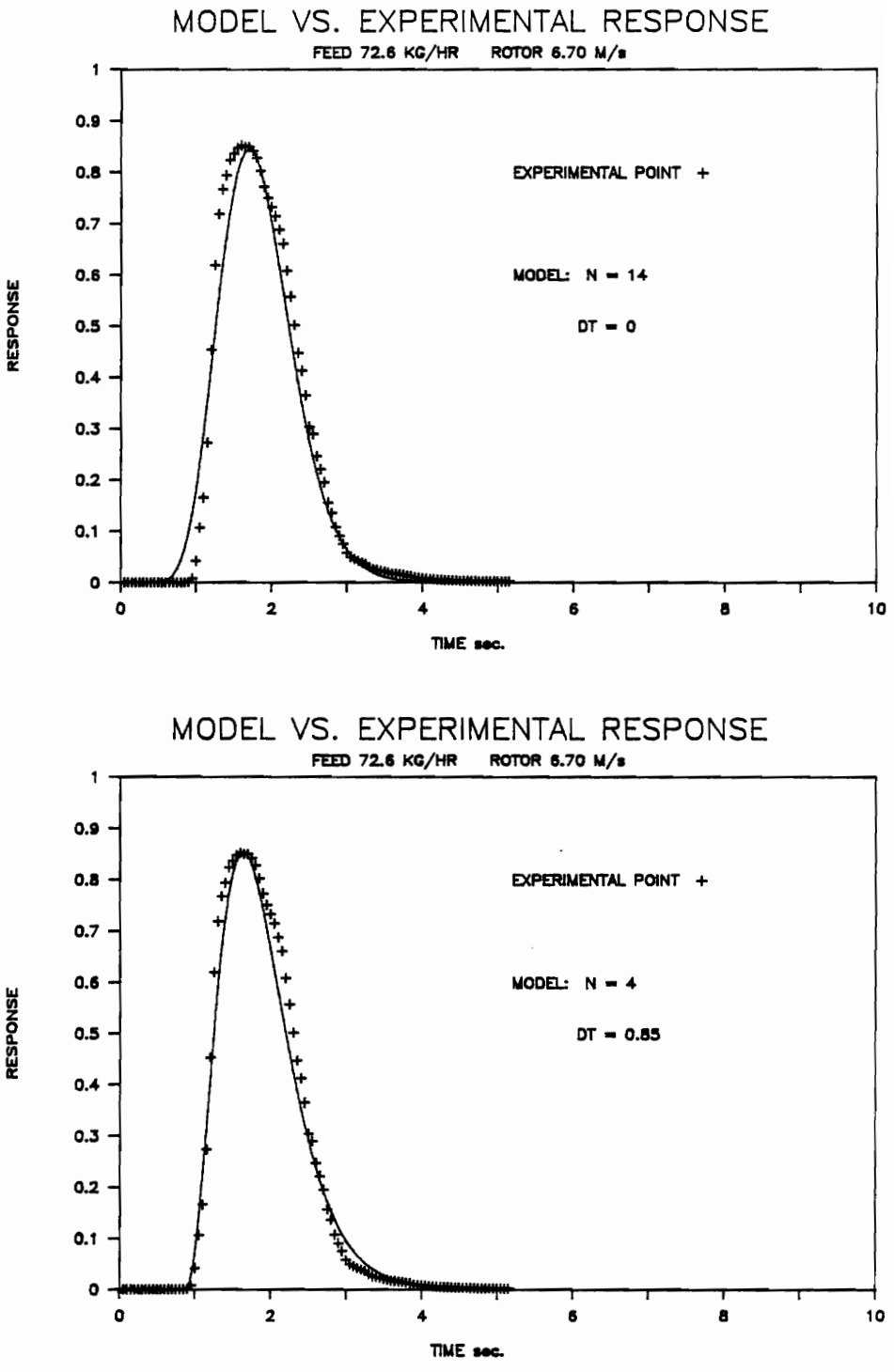
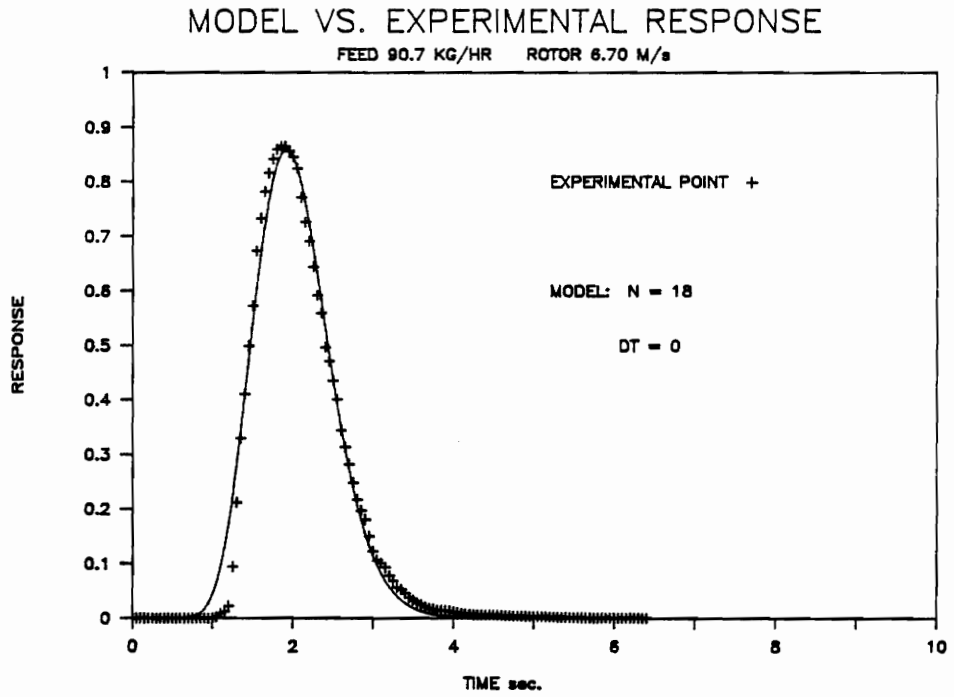


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



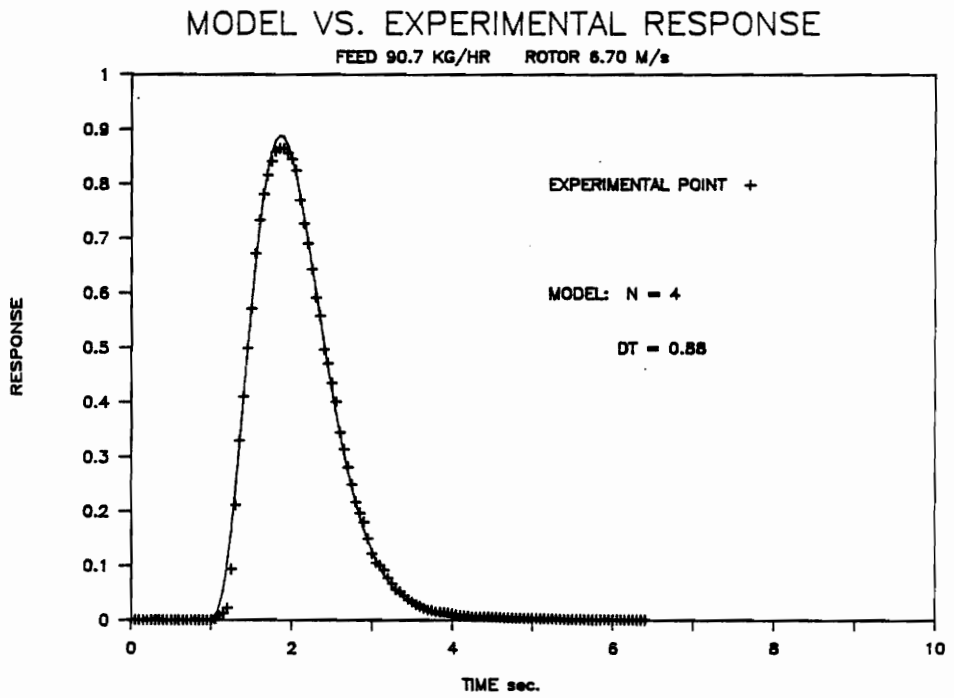
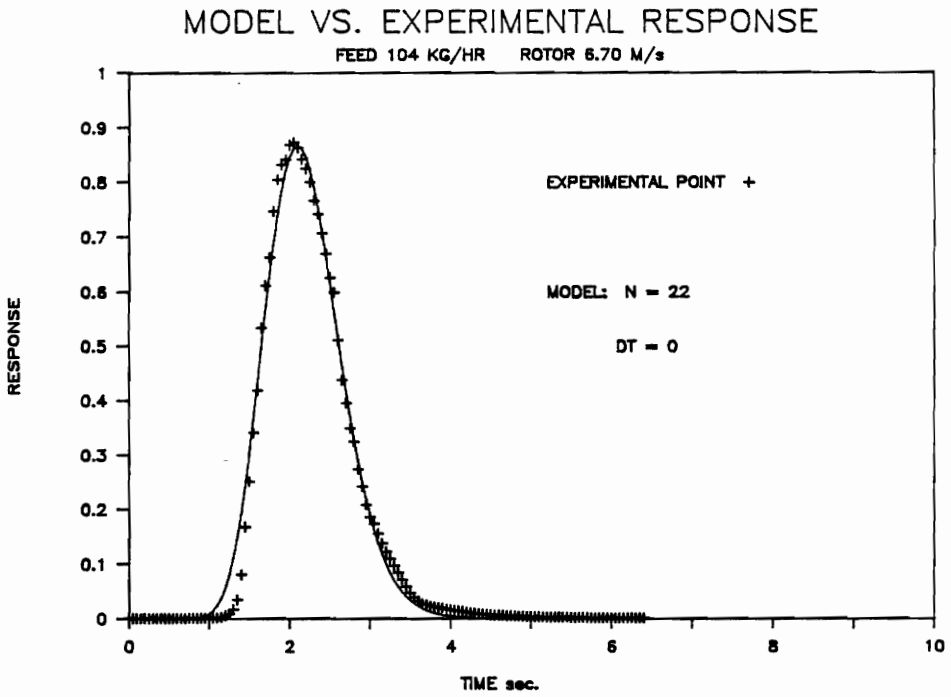


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



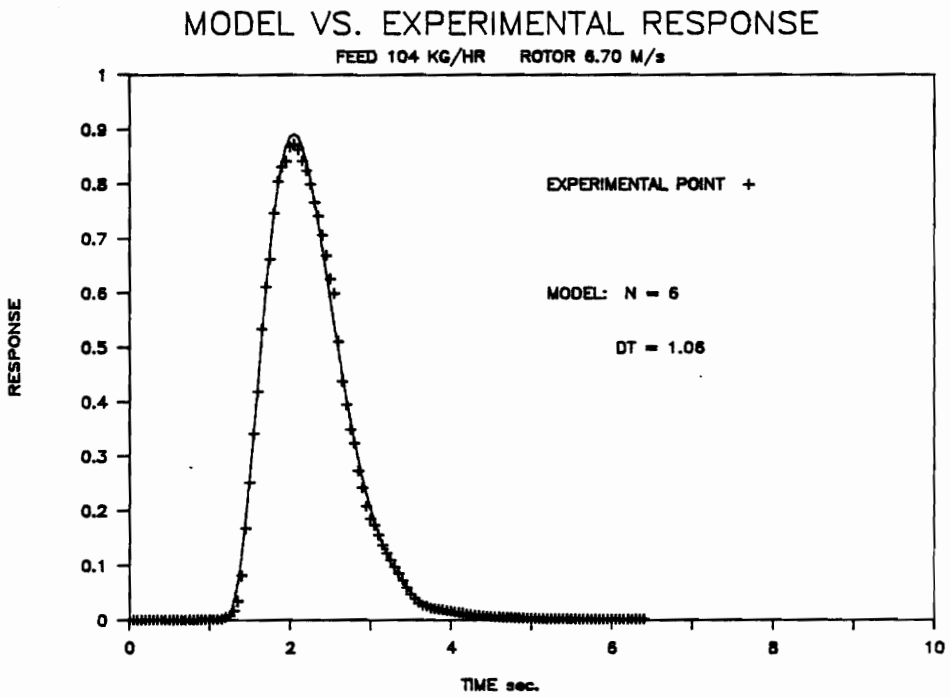


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

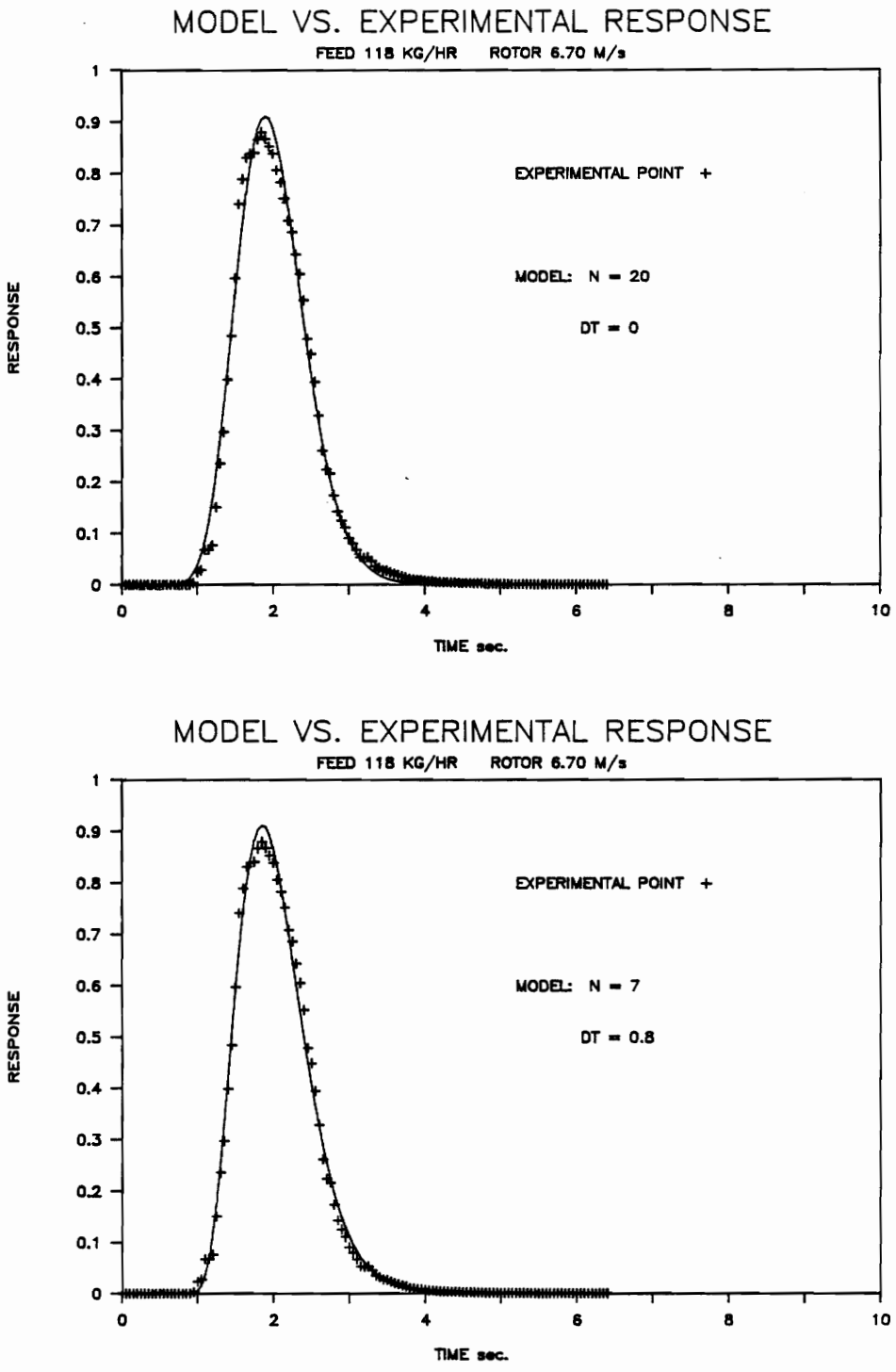


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

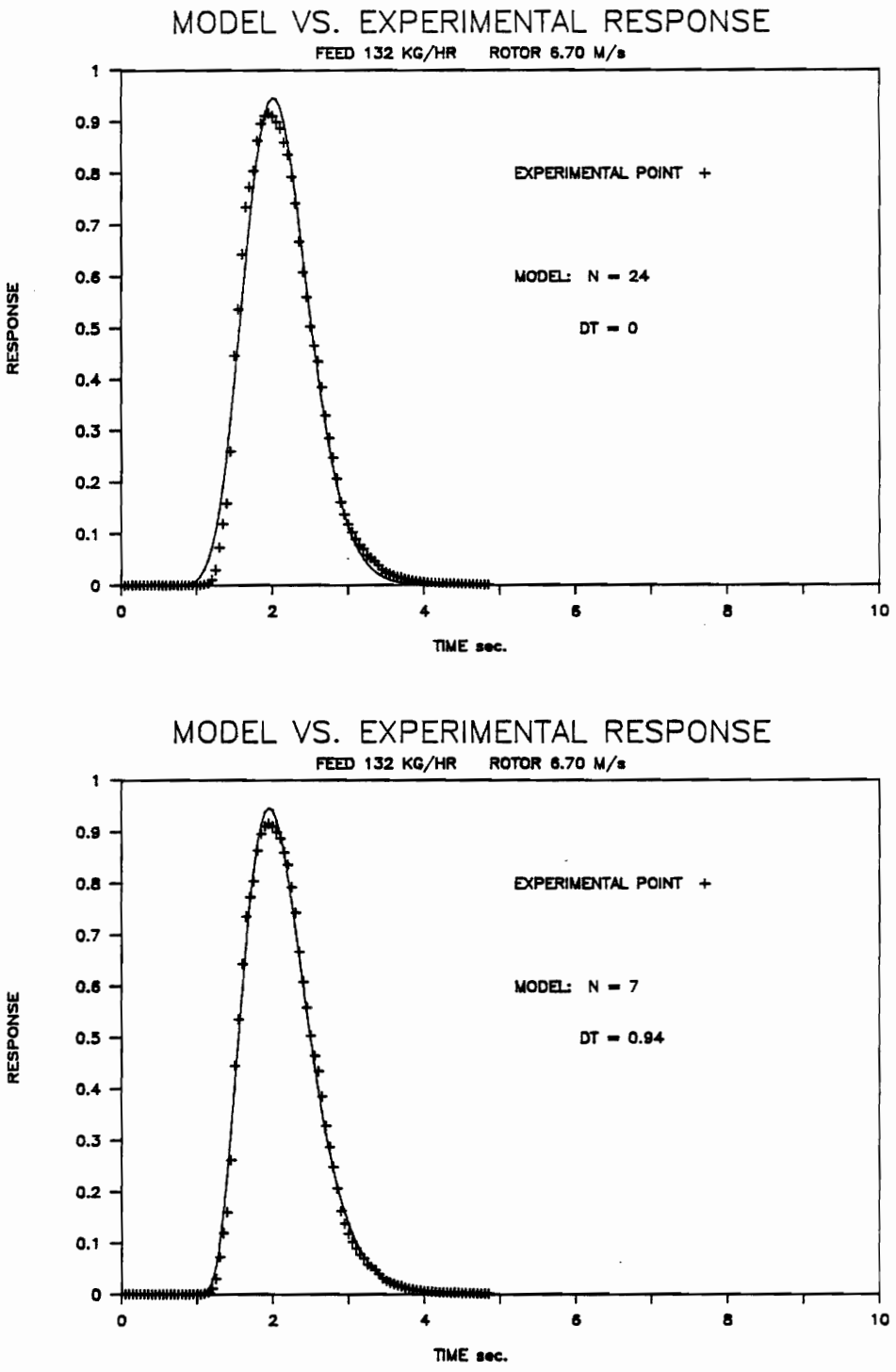


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

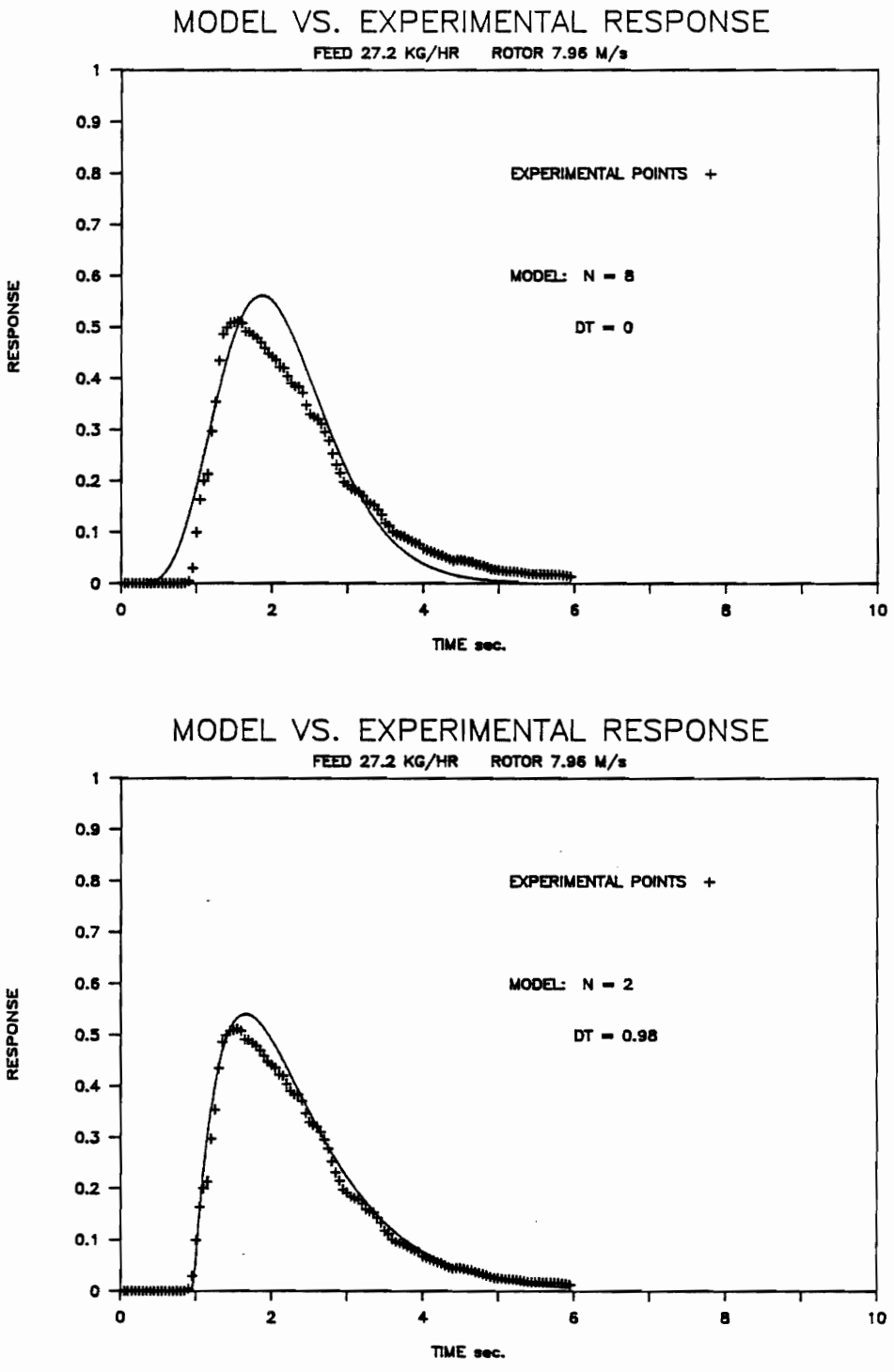


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

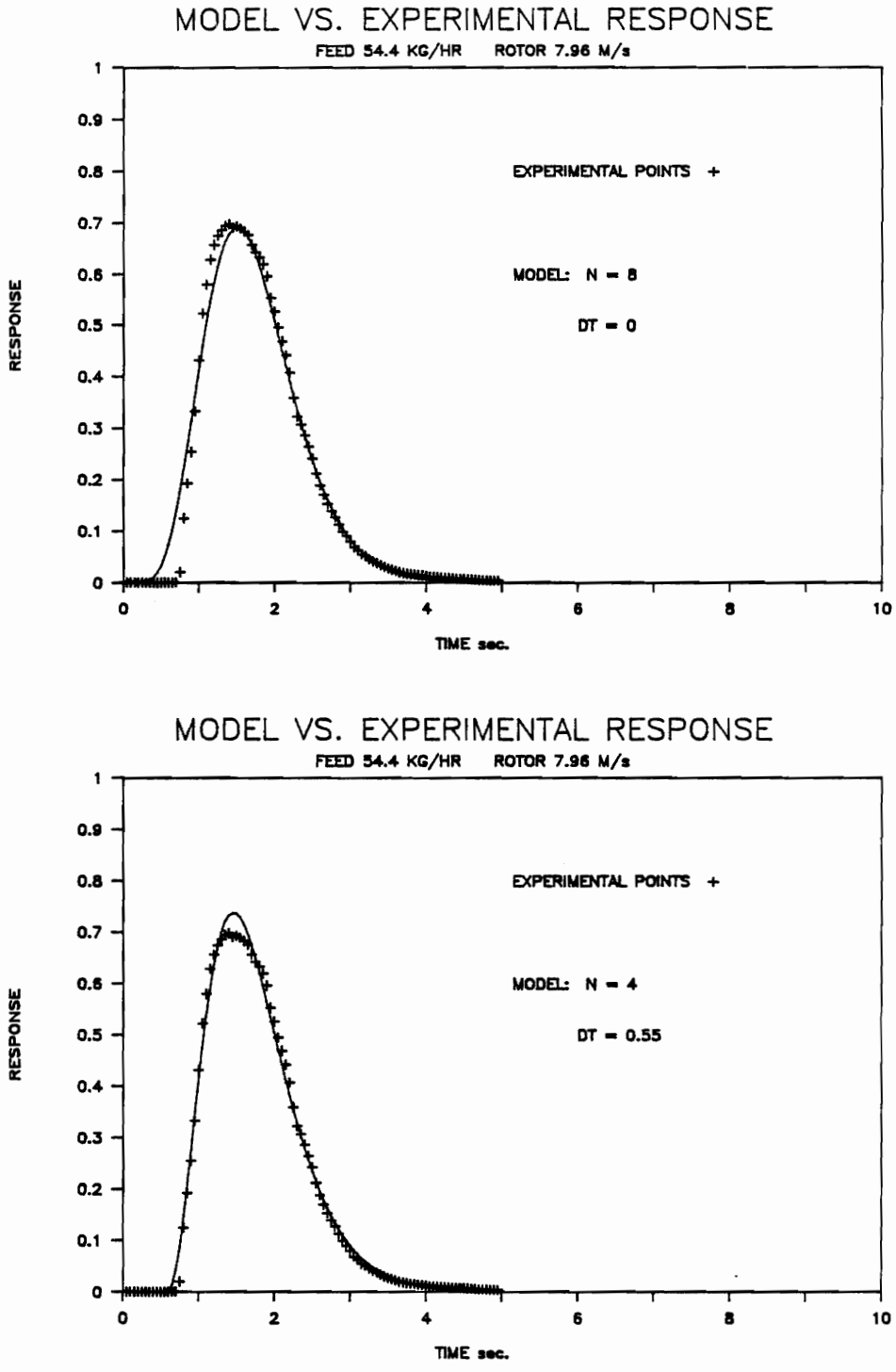


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

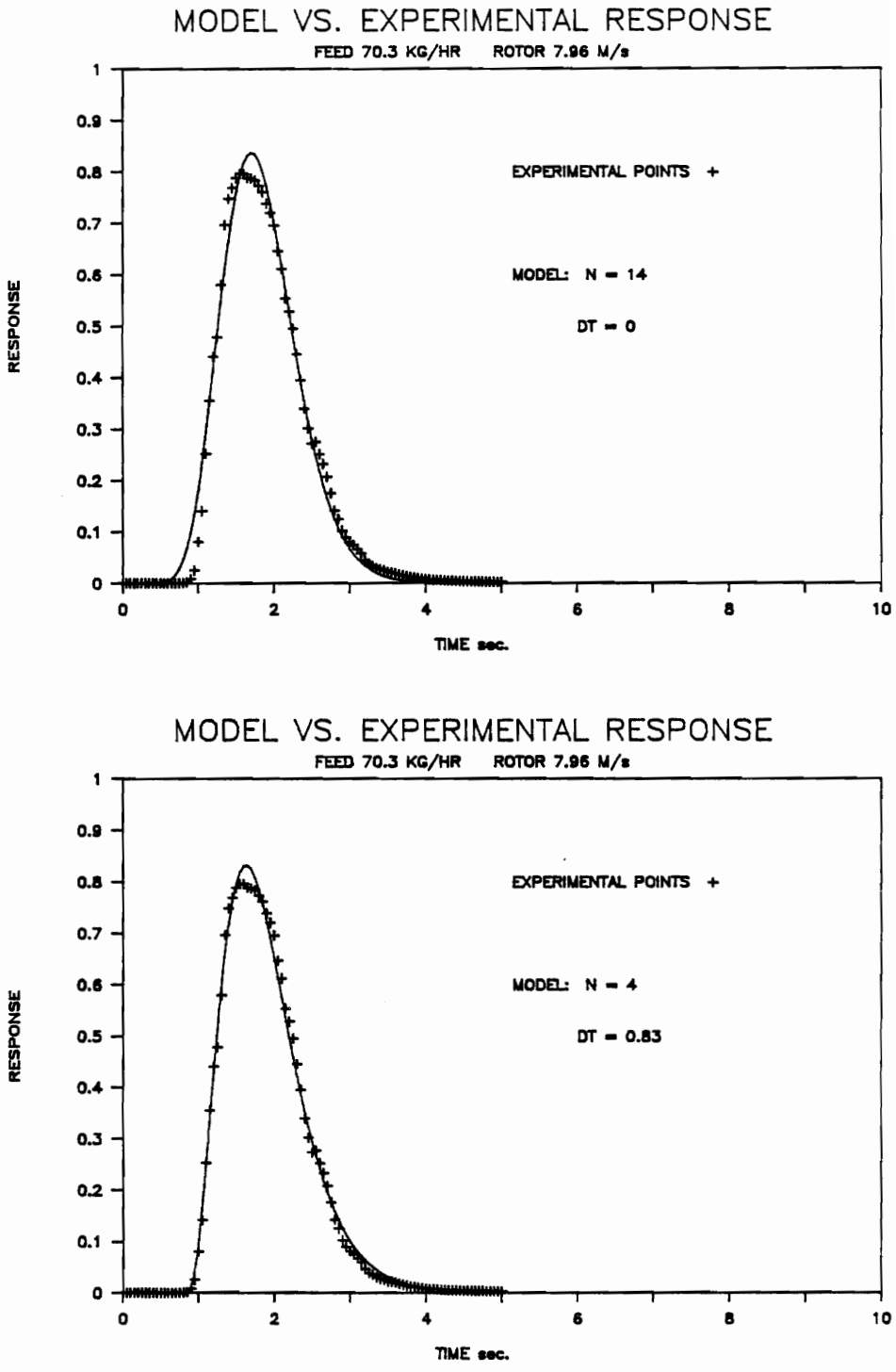


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

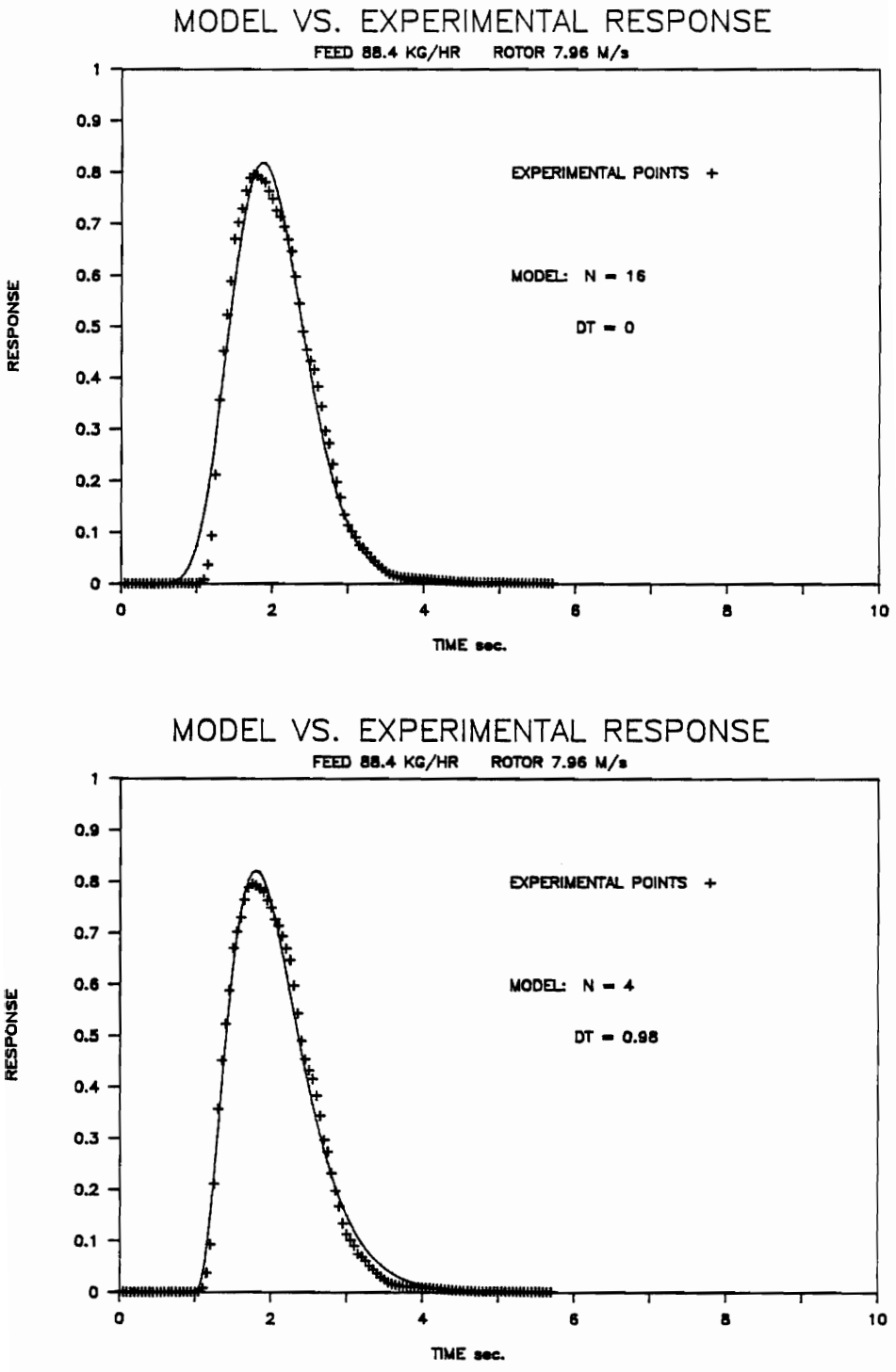
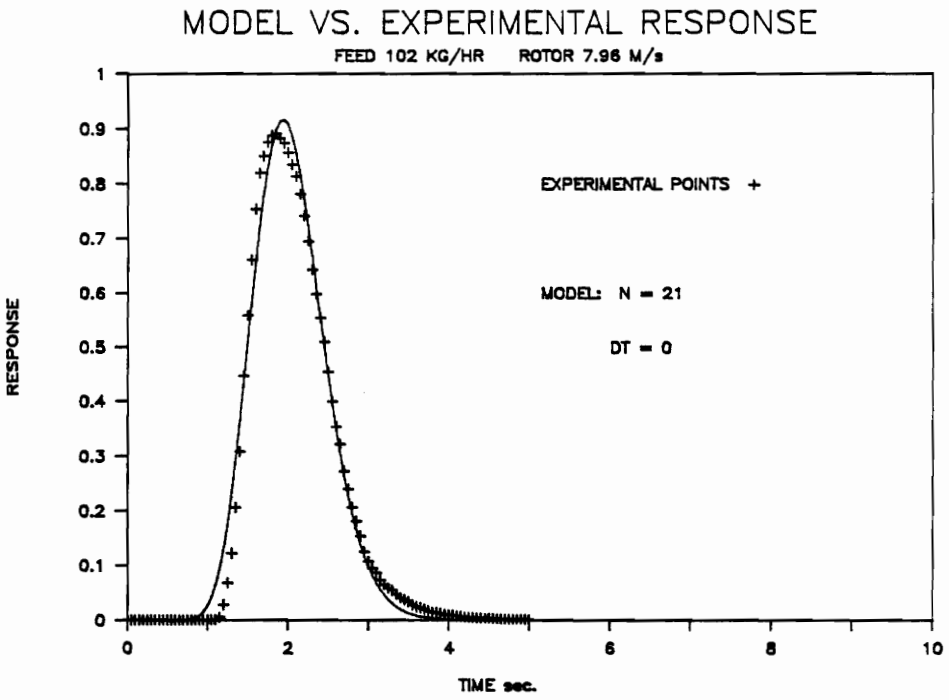


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



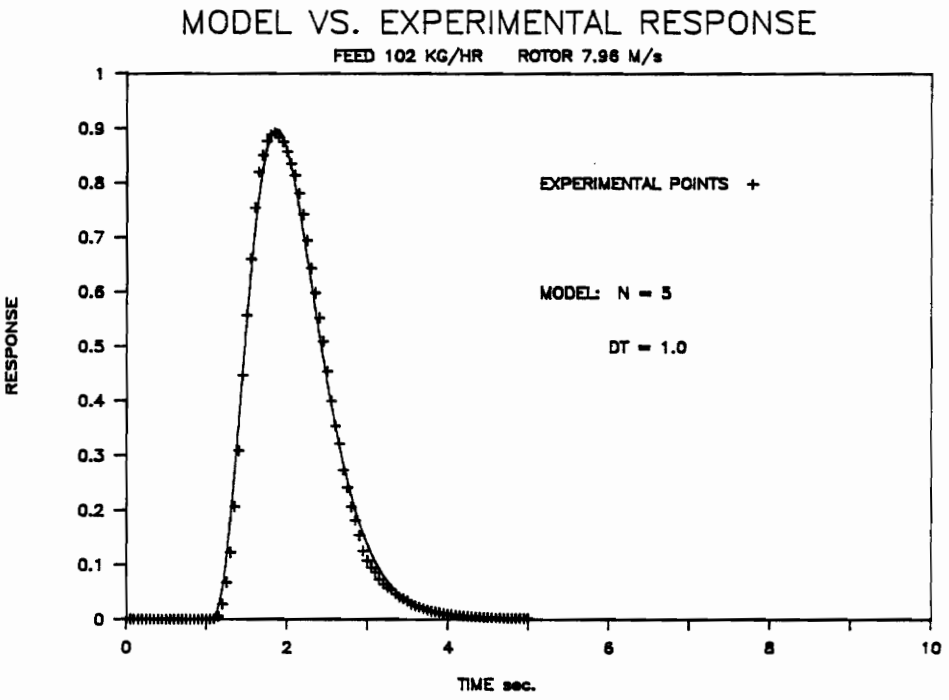


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

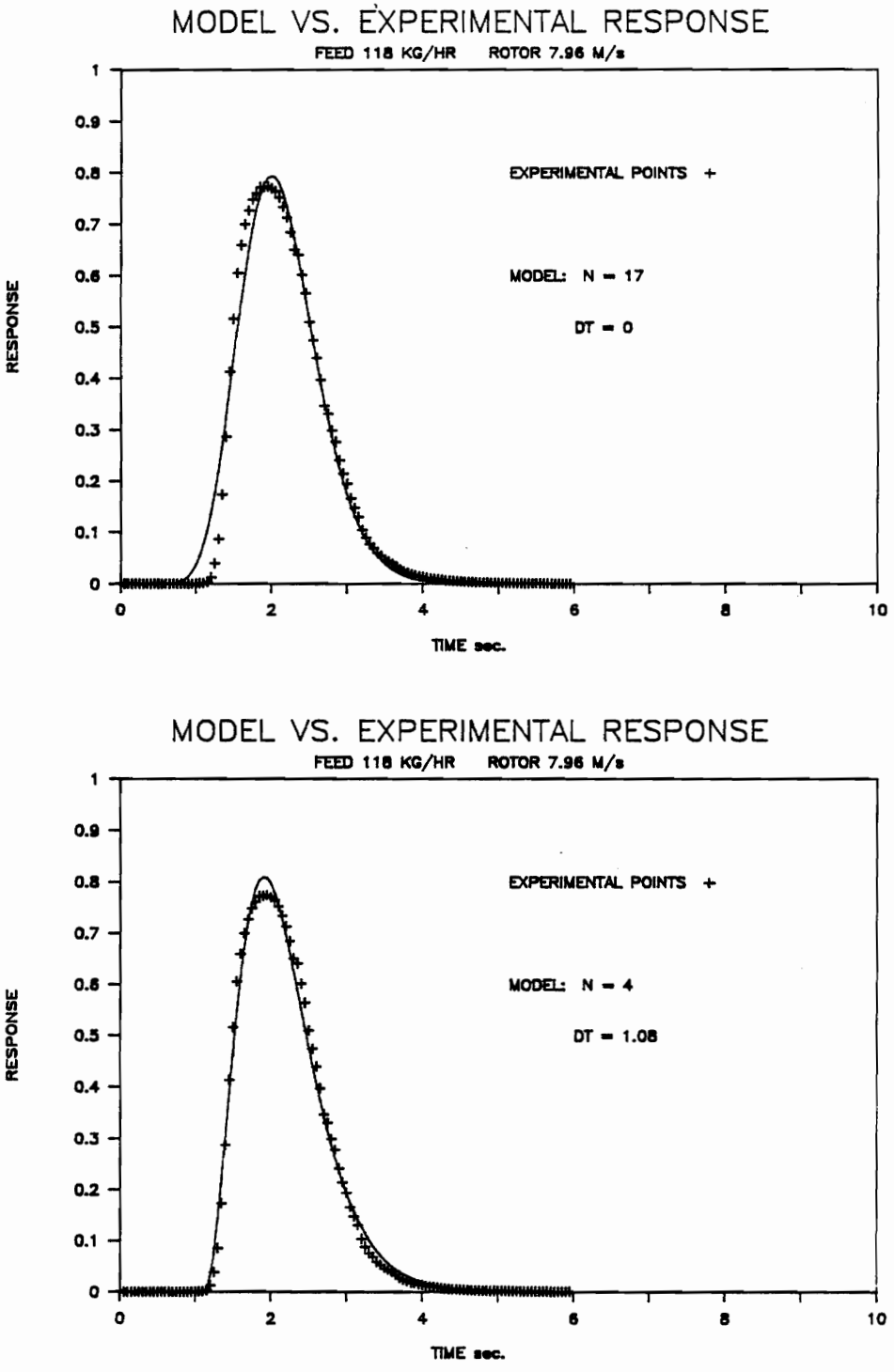


Figure A.2 Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

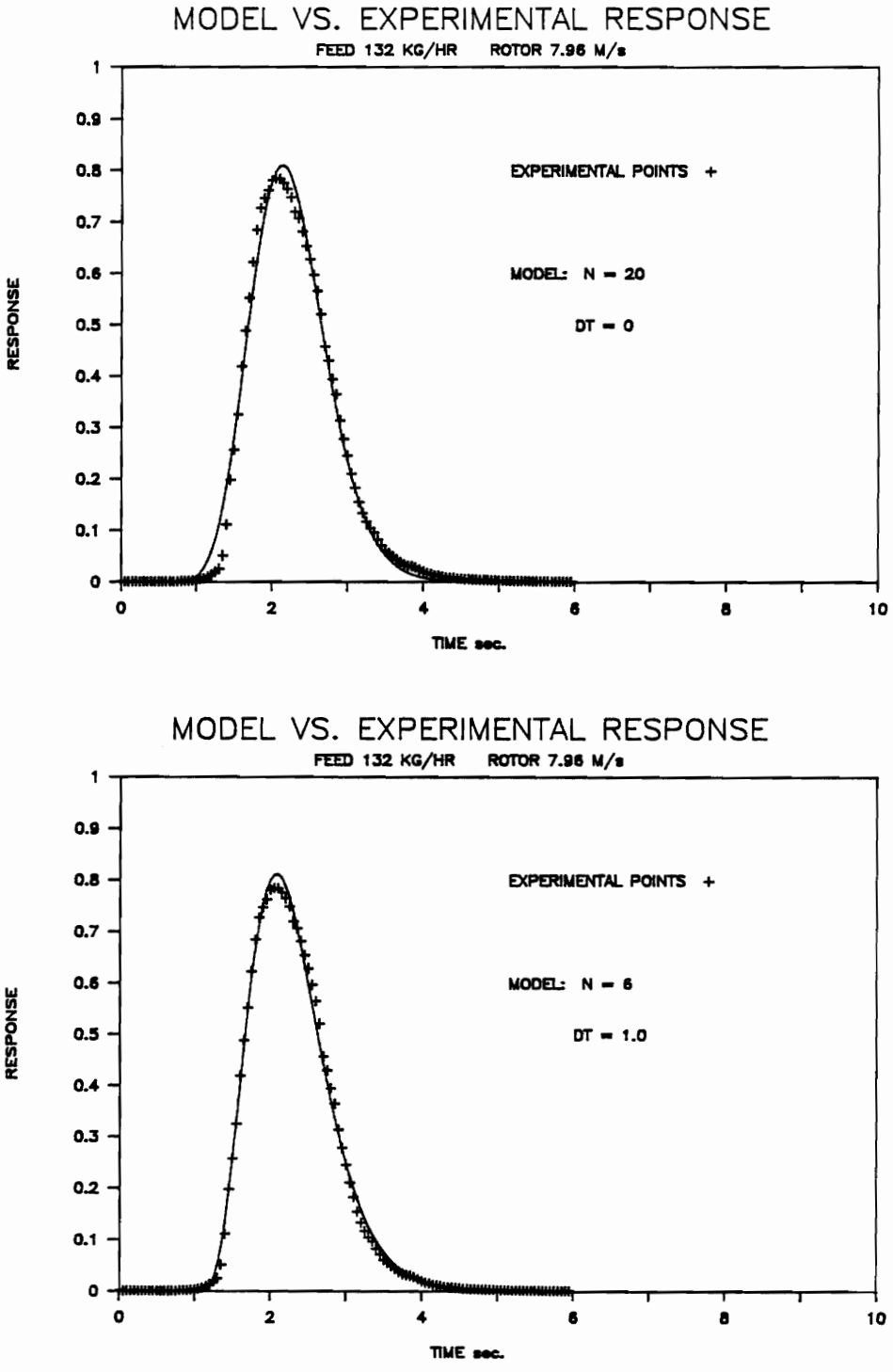


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

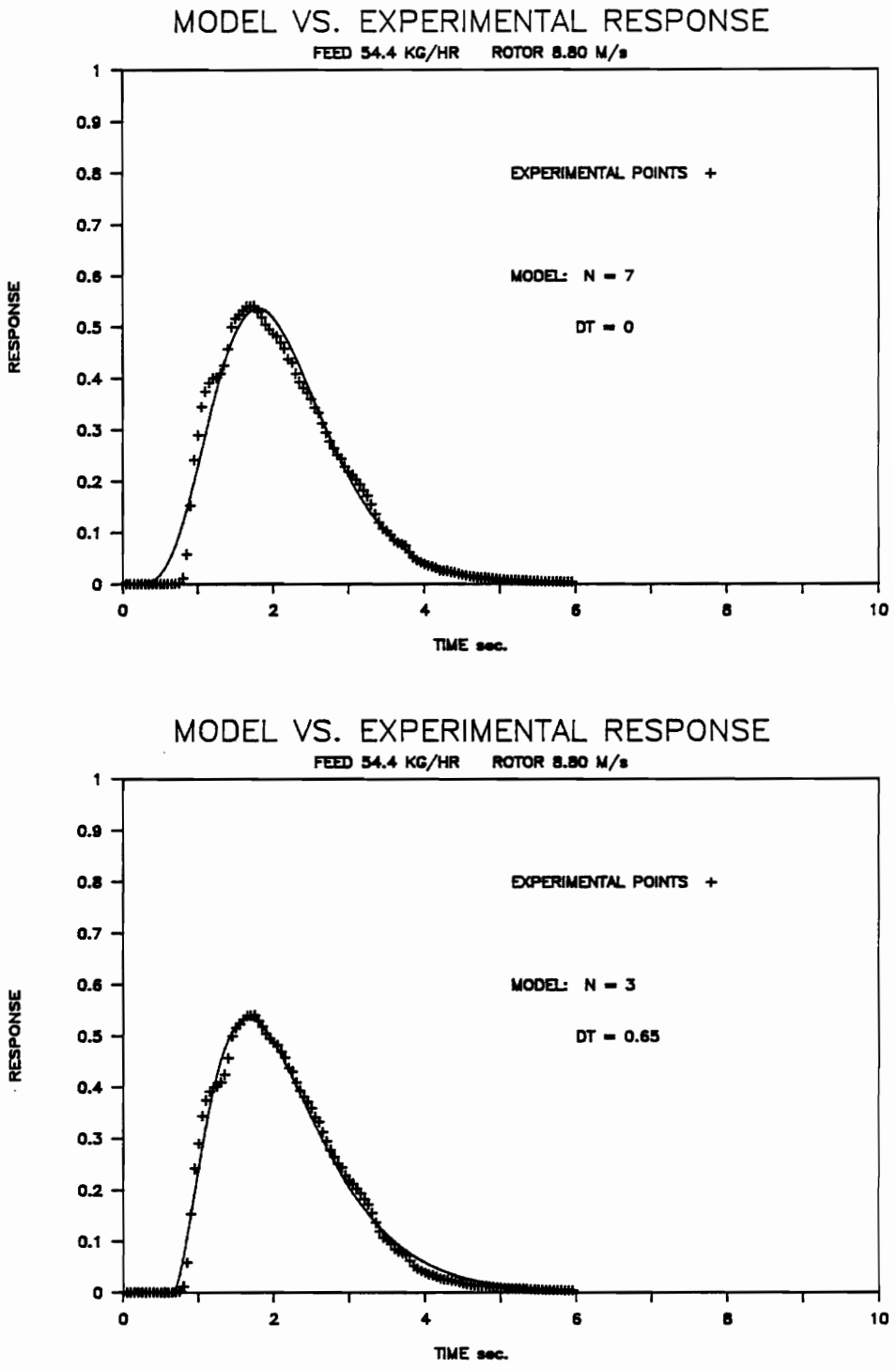
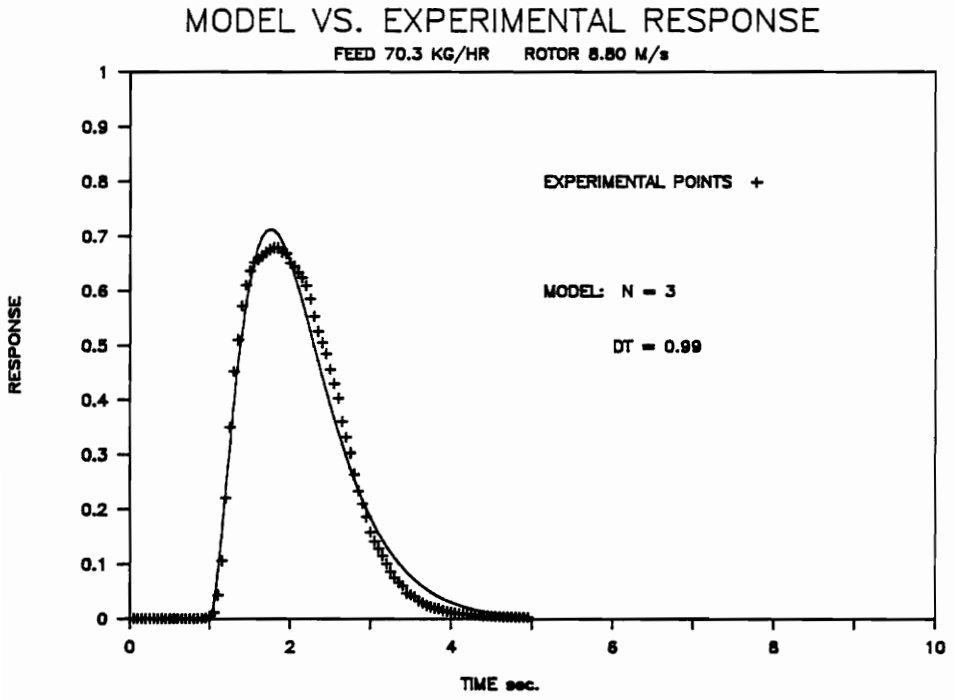


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.



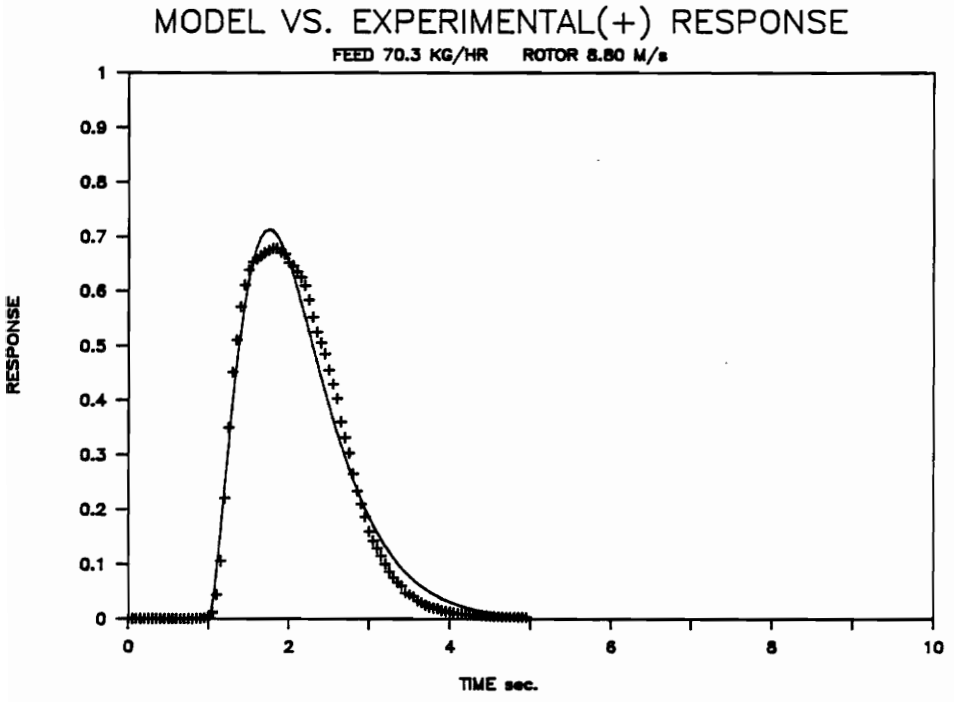


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

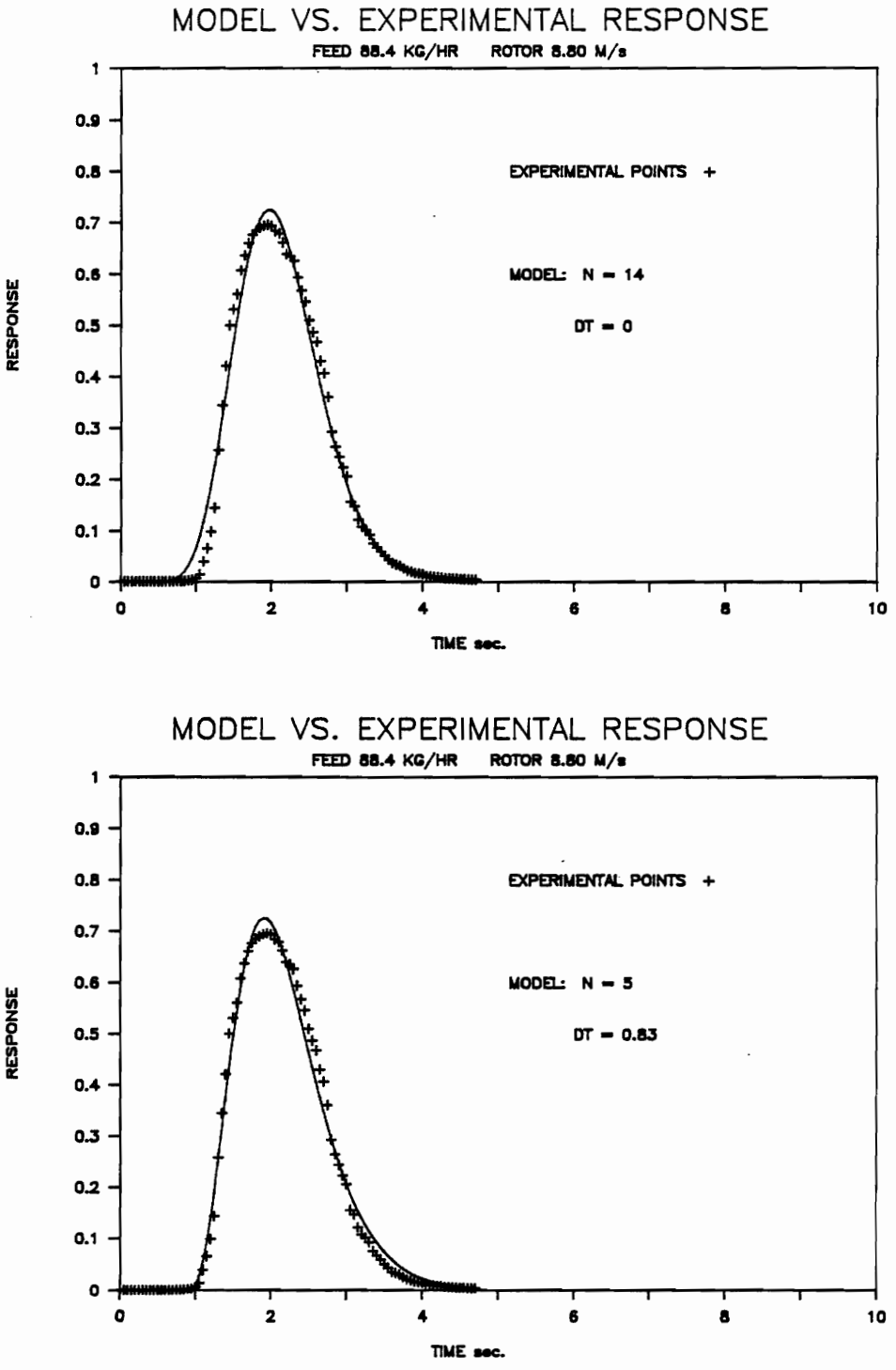


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

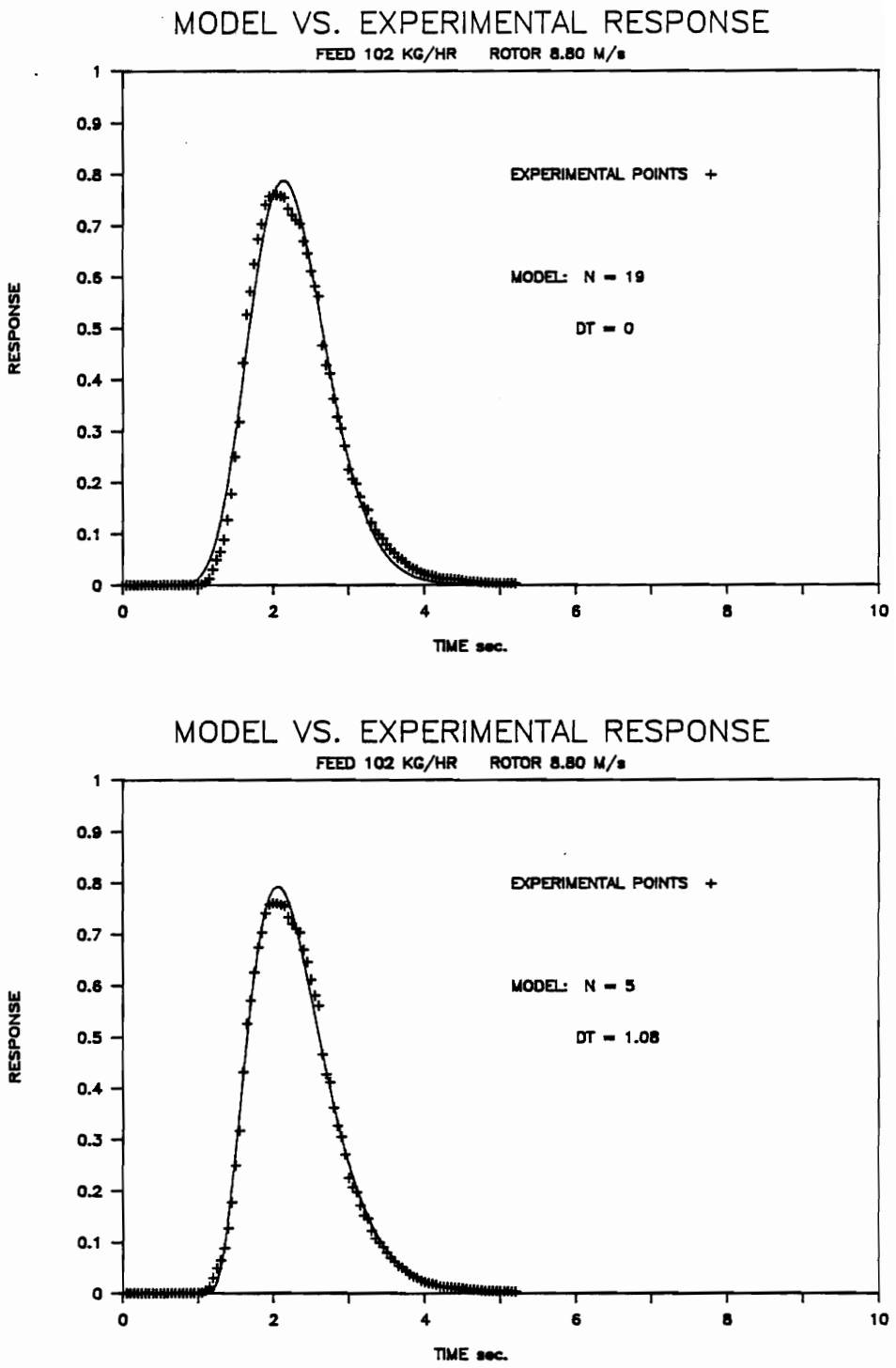


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

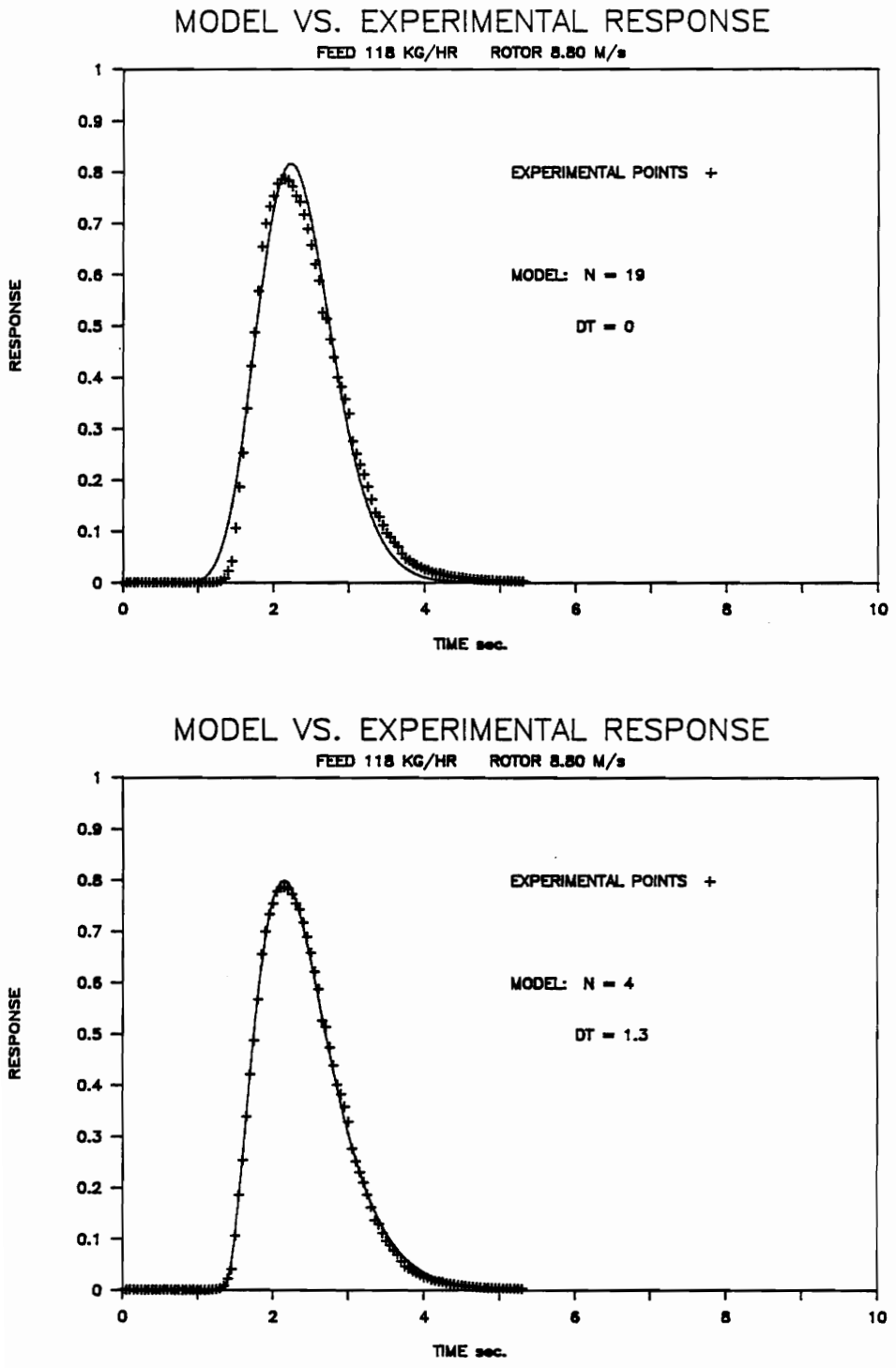


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

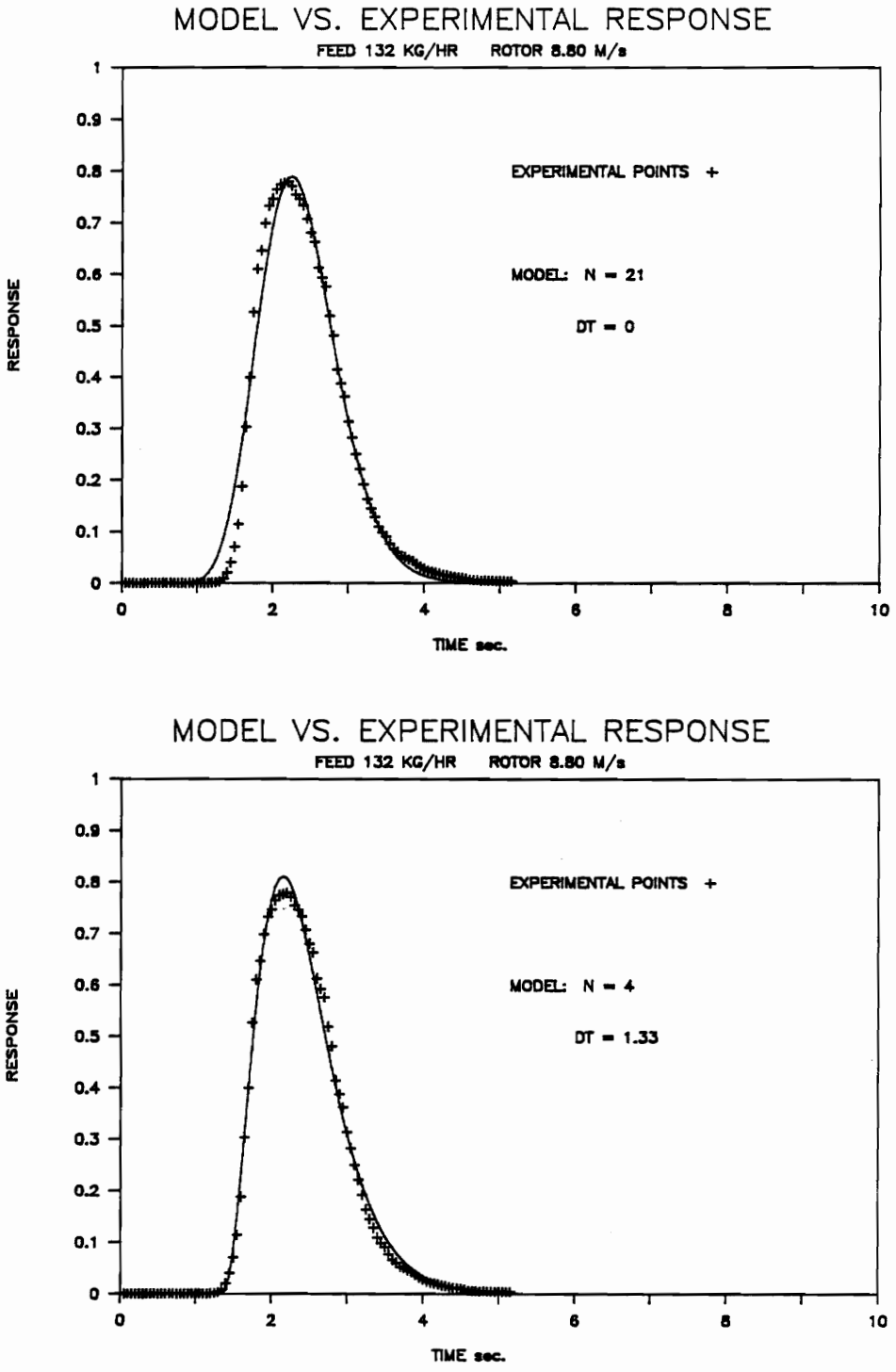


Figure A.2. Model-to-data fit with and without dead time. Model parameters are; the number of stages 'N' and the dead time 'Dt'.

APPENDIX 3

EXPERIMENTAL TRANSIENT TEMPERATURE PLOTS

The results of the transient heating temperature experiments, described in Chapter 7, are presented in the following plots.

The upper three curves are the time trajectories of the processor heating steam inlet temp. (top), outlet temp. (bottom), and average temp. (middle) values.

The lower two curves are the process fluid feed temp. (top), and discharge temp. (bottom).

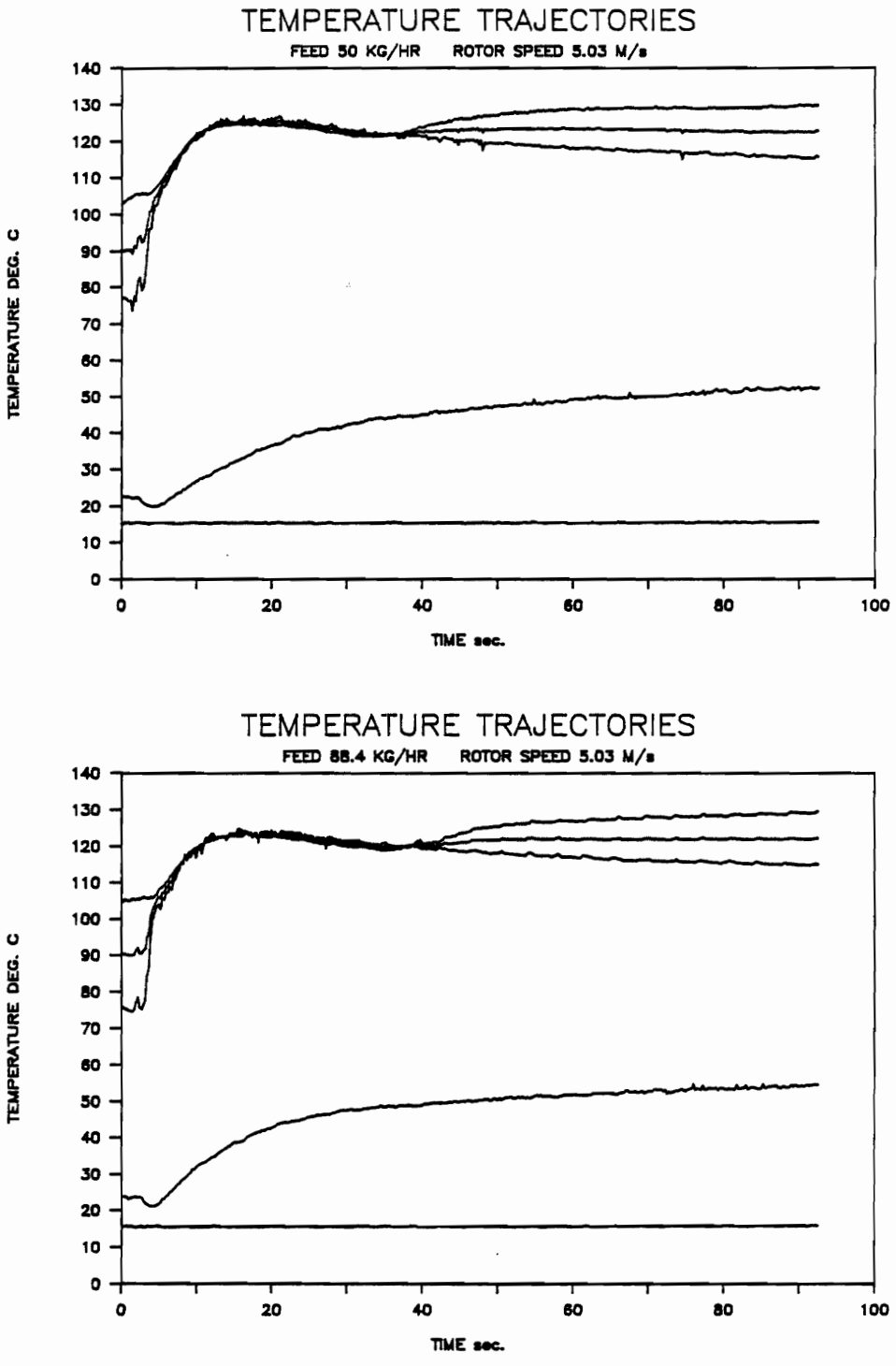


Figure A.3.1. Thin-film processor experimental transient temperature response.

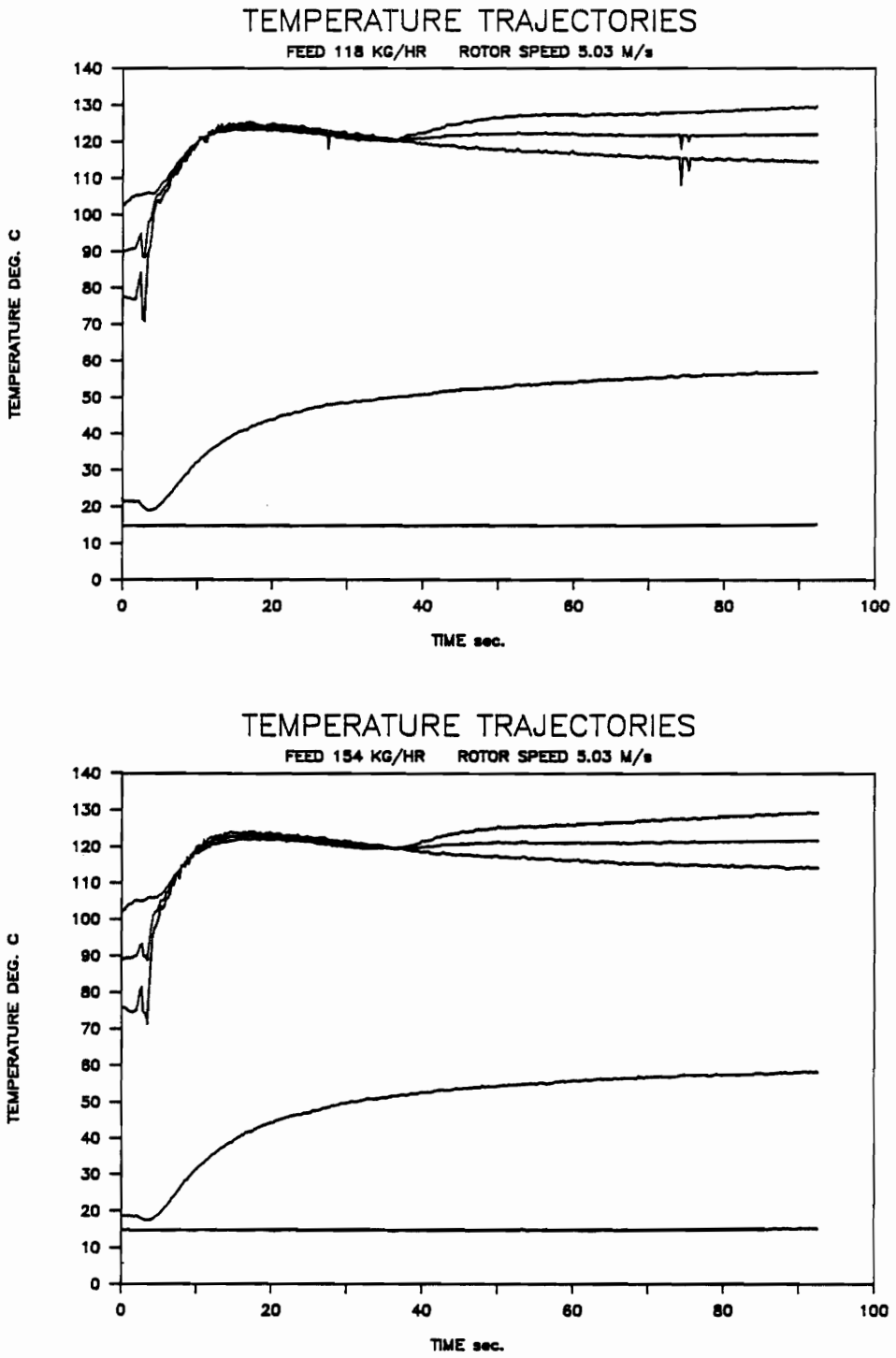


Figure A.3.2. Thin-film processor experimental transient temperature response.

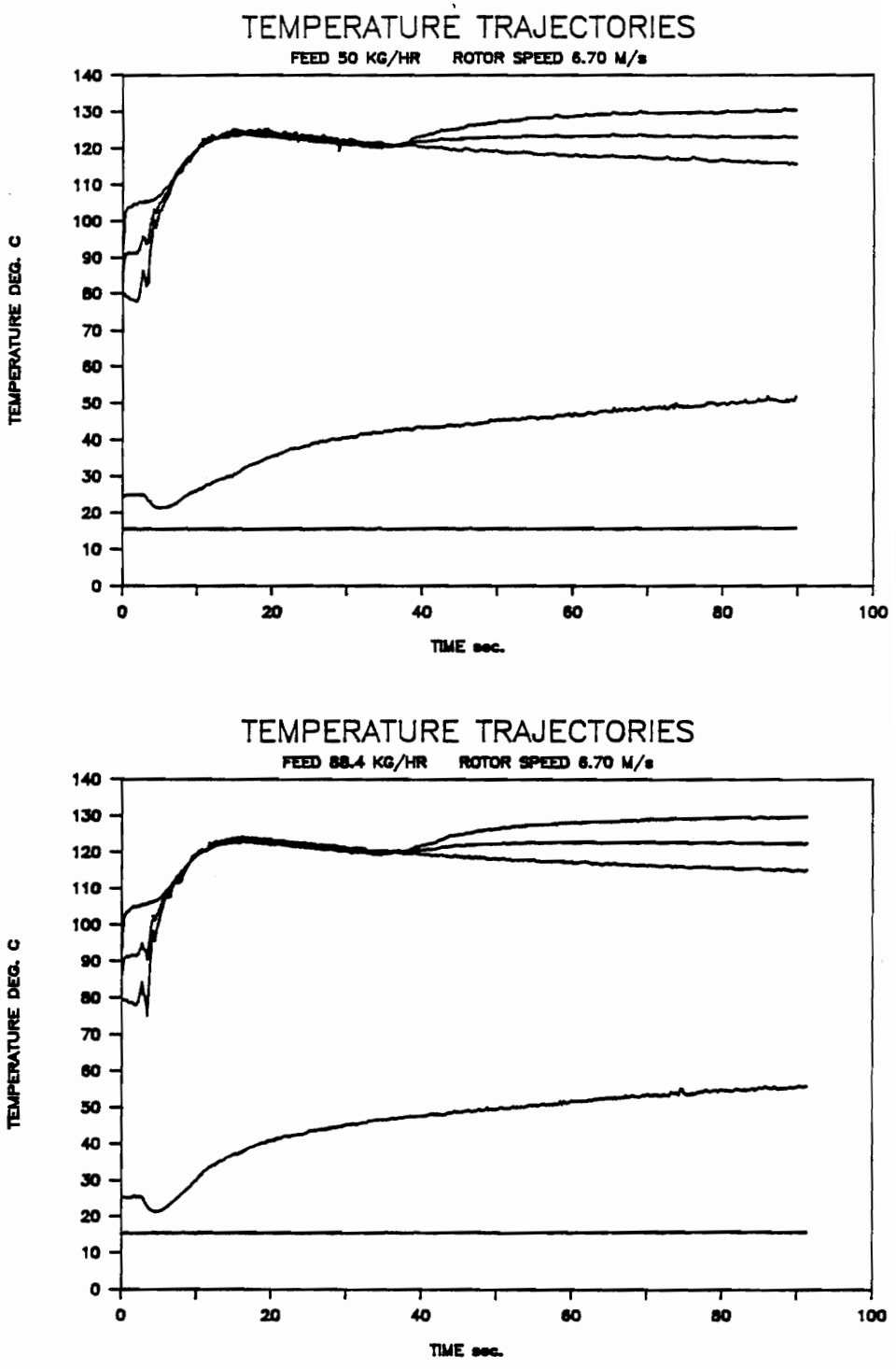


Figure A.3.3. Thin-film processor experimental transient temperature response.

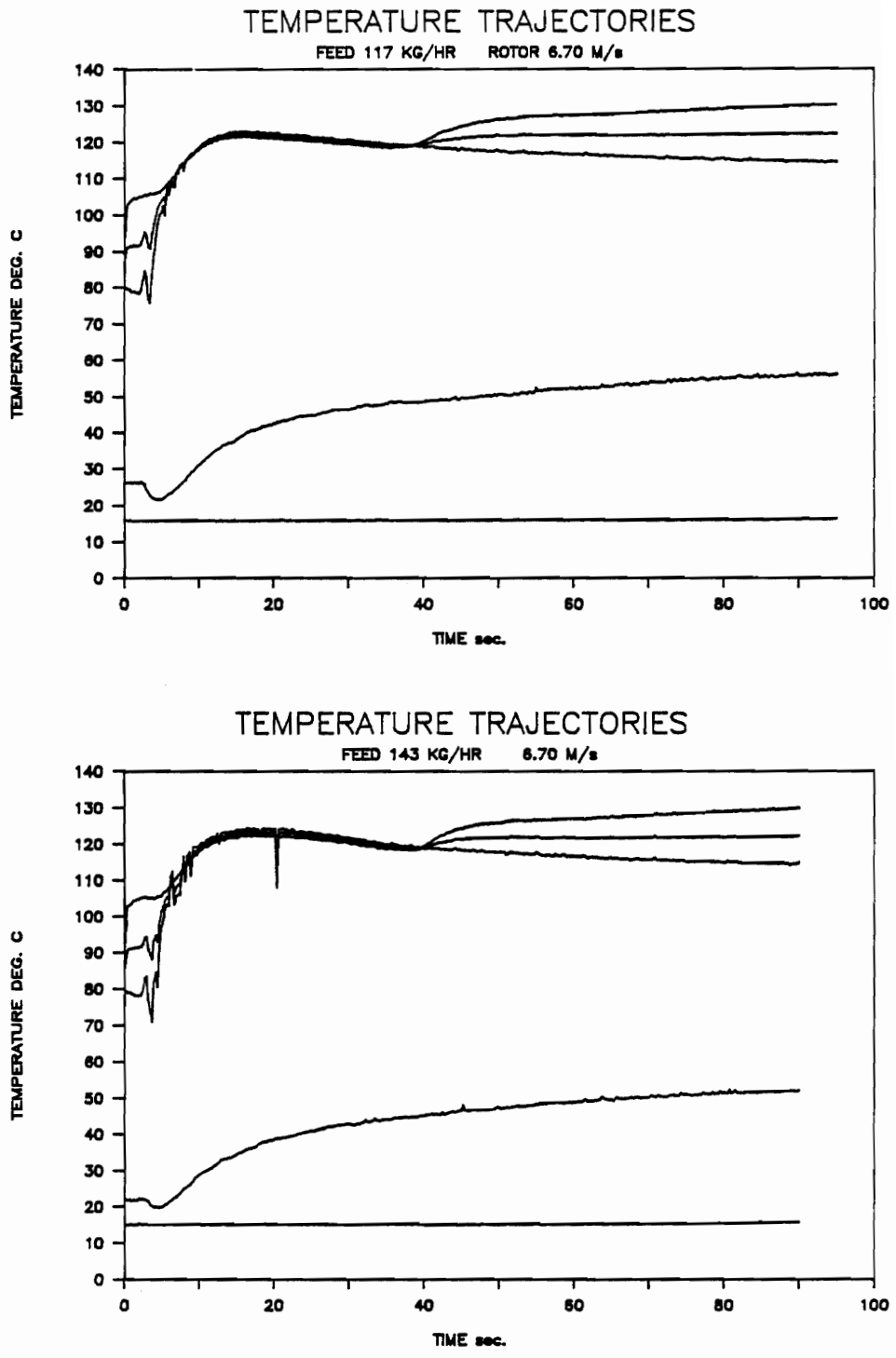


Figure A.3.4. Thin-film processor experimental transient temperature response.

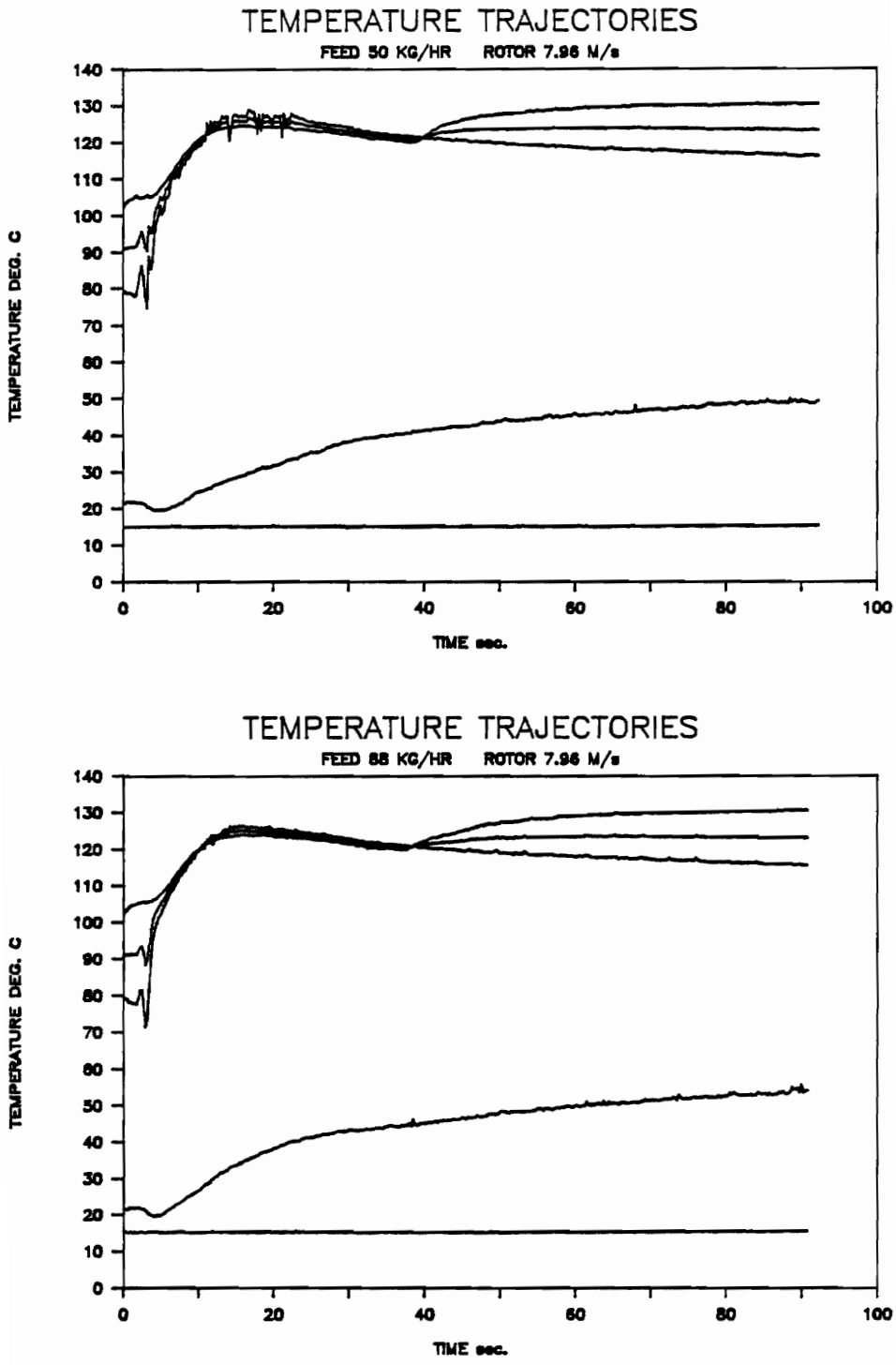


Figure A.3.5. Thin-film processor experimental transient temperature response.

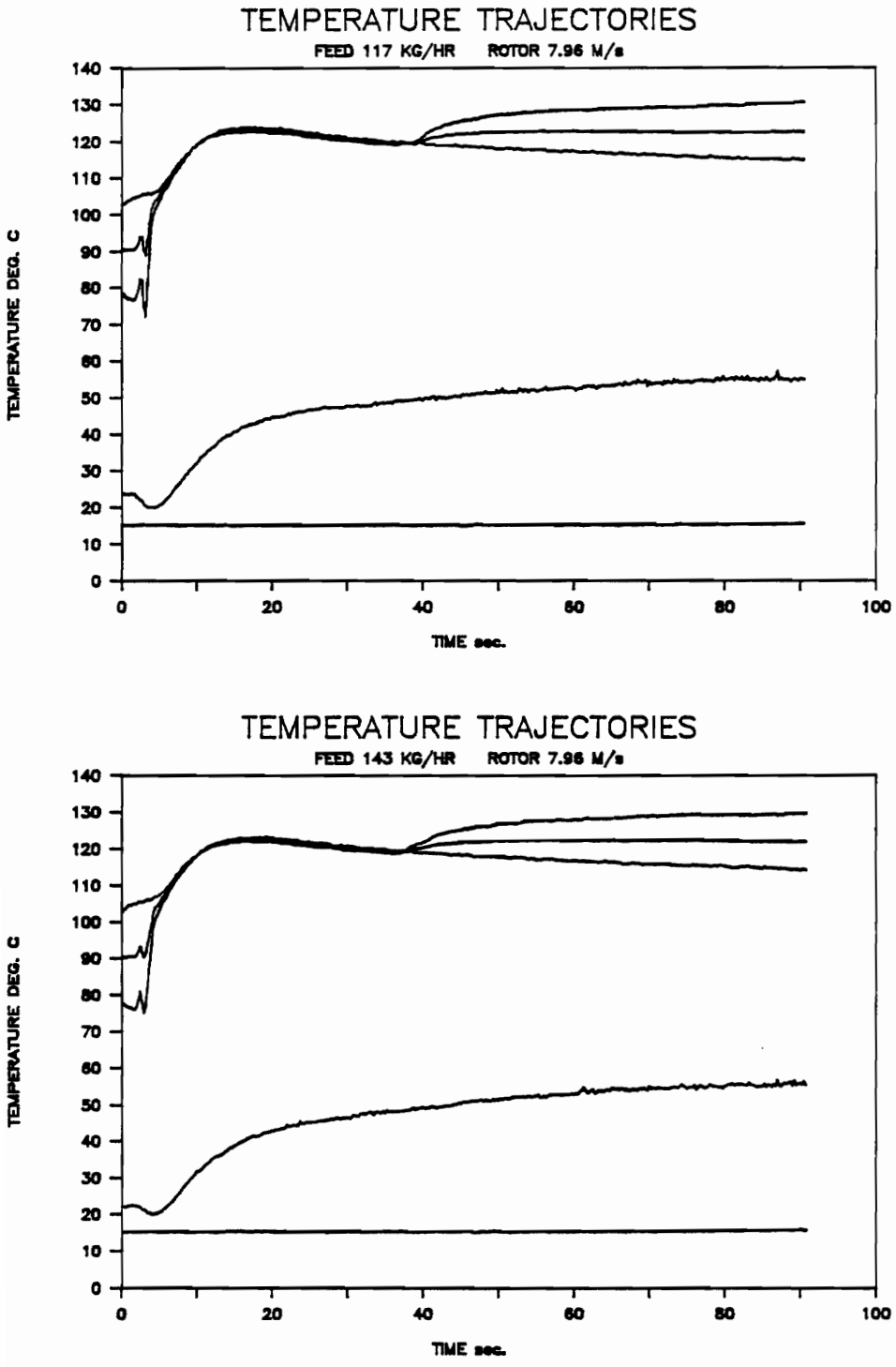


Figure A.3.6. Thin-film processor experimental transient temperature response.

APPENDIX 4

TRANSIENT TEMPERATURE SIMULATION PLOTS

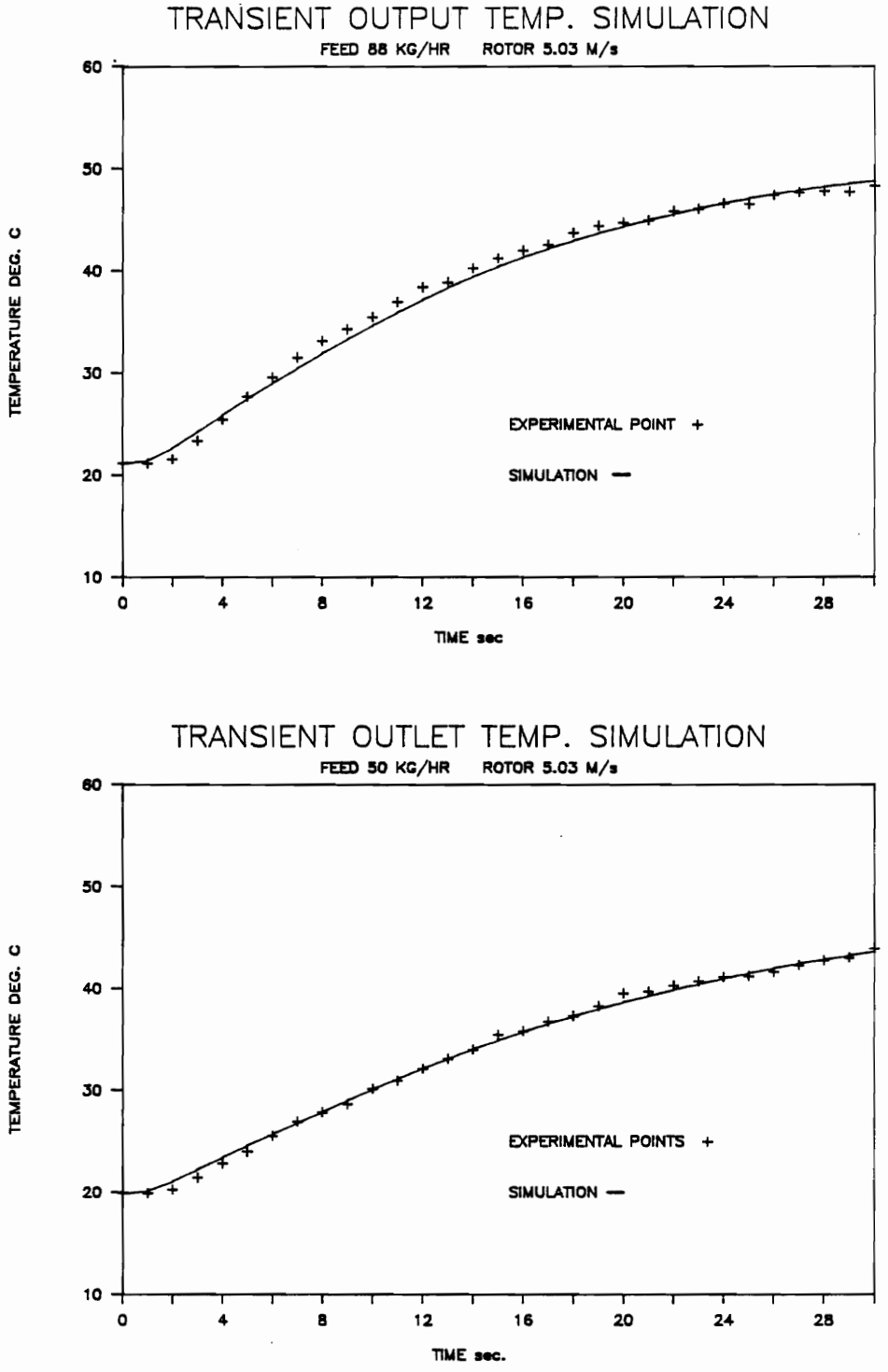


Figure A.4.1. Simulation-to-data fit of the processor transient bottom discharge temperature.

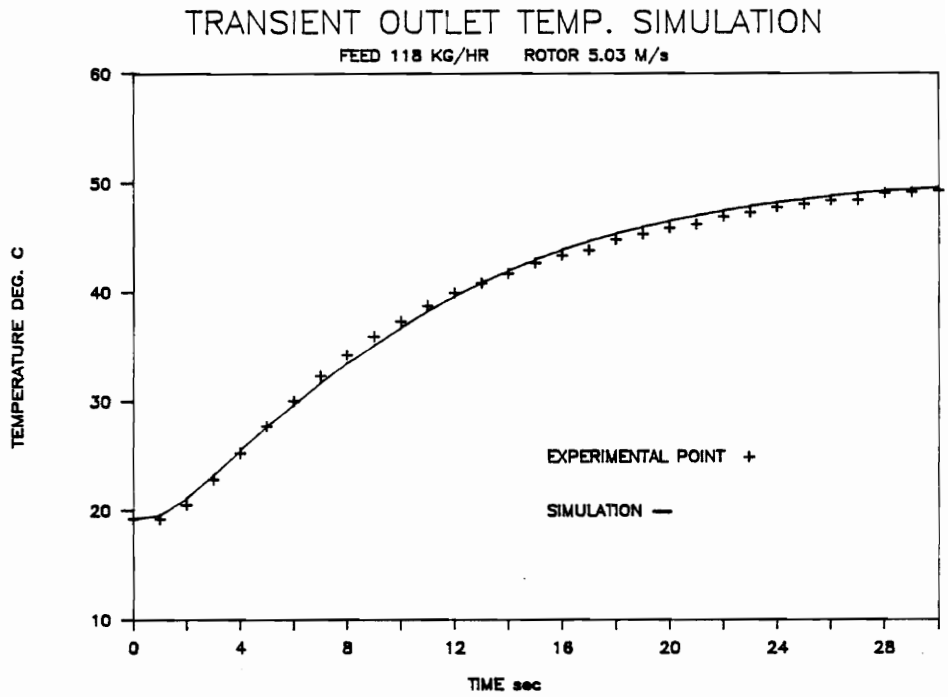
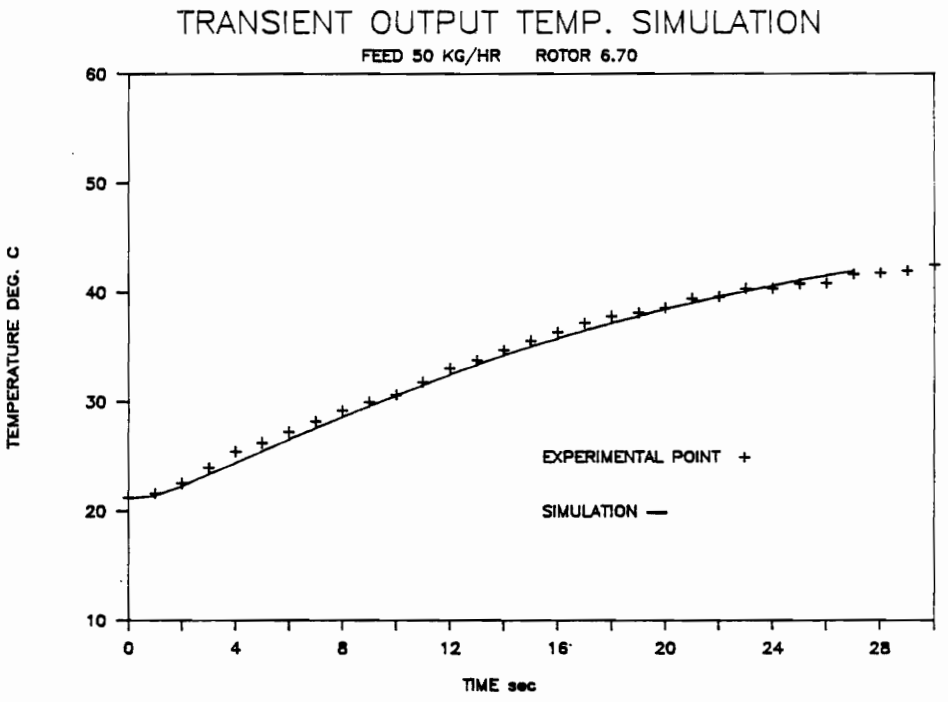


Figure A.4.2. Simulation-to-data fit of the processor transient bottom discharge temperature.



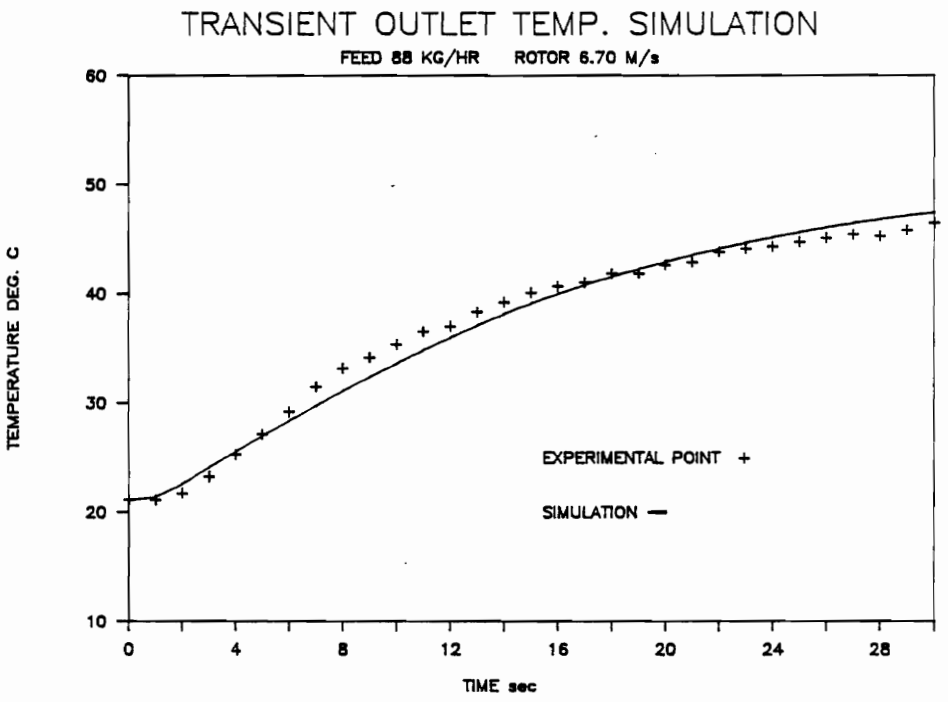
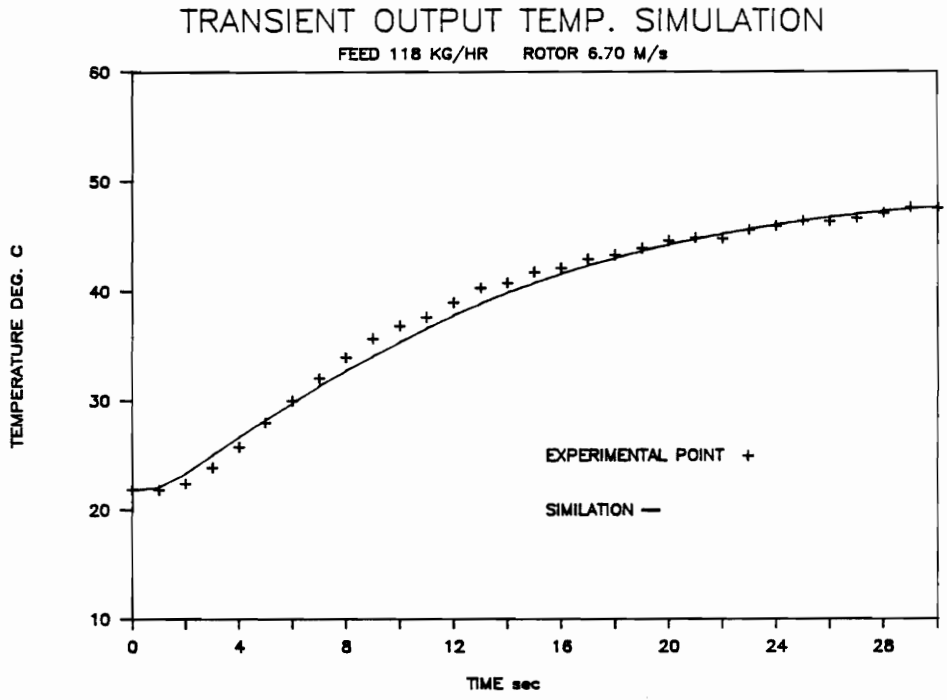


Figure A.4.3. Simulation-to-data fit of the processor transient bottom discharge temperature.



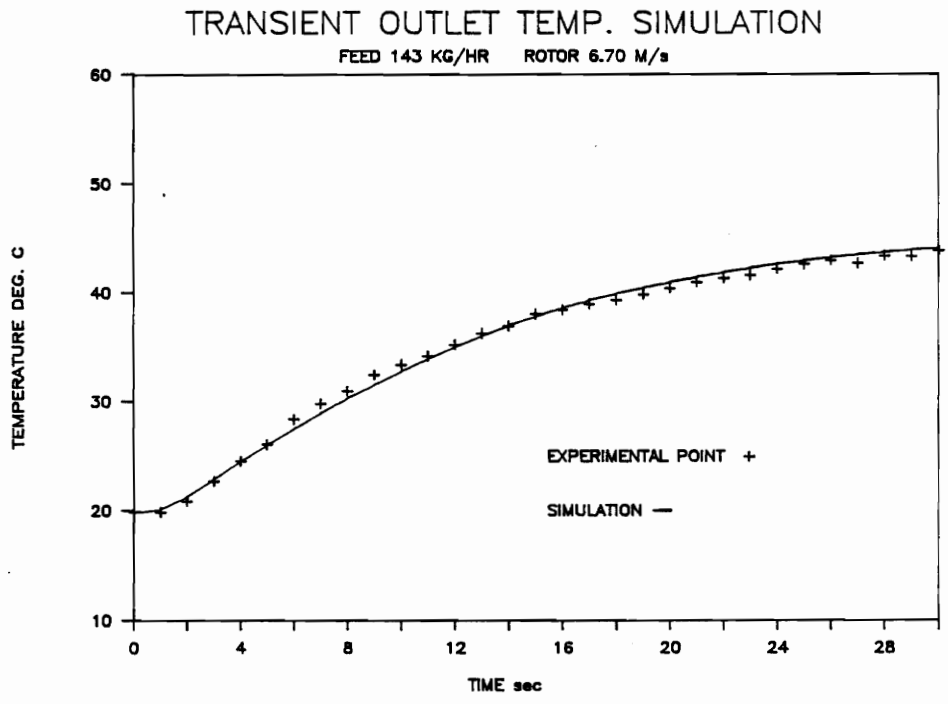


Figure A.4.4. Simulation-to-data fit of the processor transient bottom discharge temperature.

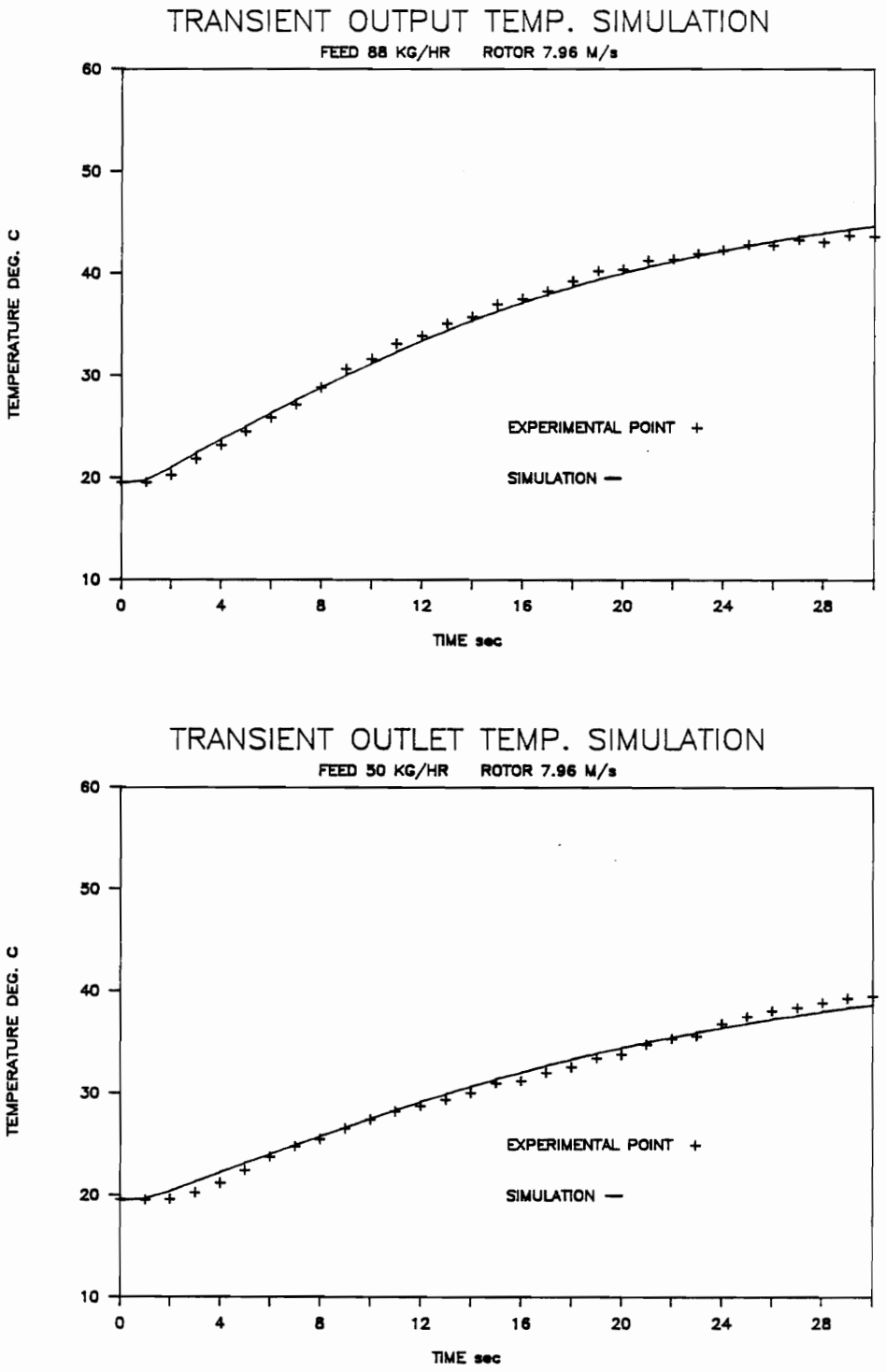


Figure A.4.5. Simulation-to-data fit of the processor transient bottom discharge temperature.

APPENDIX 5

DETAILS OF THE CONDUCTIVITY TRANSDUCER

Conductivity Transducer Circuit Details

The following paragraphs refer to the circuit diagram shown in Figure A.5.1.

The 1000 Hz input sine wave is generated by an integrated circuit function generator.

Potentiometer A1 sets the amplitude of the sinusoidal input function to the transducer amplifier U2. Capacitor C1 and resistor R1 form a high pass filter (cutoff frequency 100 Hz) to eliminate signal baseline drift from the function generator. Resistor R2 is the variable impedance of the conductivity probe immersed in the process fluid. Potentiometer A2 scales transducer amplification such that the lowest anticipated probe impedance will not result in a gain large enough to saturate the amplifier.

Amplifier U3 is configured as an active, second order, band pass filter:

low frequency cutoff - 800 Hz

high frequency cutoff - 1100 Hz

circuit Q - 1.57

circuit gain - 1.0

Amplifiers U4 and U5 are configured in an absolute value, or precision rectifier, circuit. For negative inputs, U4 behaves as a voltage follower through the lower diode. For positive inputs, it behaves as an inverting, unity gain, amplifier through the upper diode. Amplifier U5

supplies a constant output impedance to the rectifier circuit to insure the current through the diodes exceeds the value for threshold conduction.

Resistors R11, R12, and capacitor C5 form a low pass filter that serves as the input stage to the signal integrator. The integrator, U6, is 'zeroed' by an externally adjustable potentiometer, A3, which balances the inverting and non-inverting inputs when a null signal is presented to the integrating amplifier. The amplifier gain is controlled by the external potentiometer, A4.

The output stage, U7, inverts the (negative) integrator output, and provides impedance buffering for the final output signal.

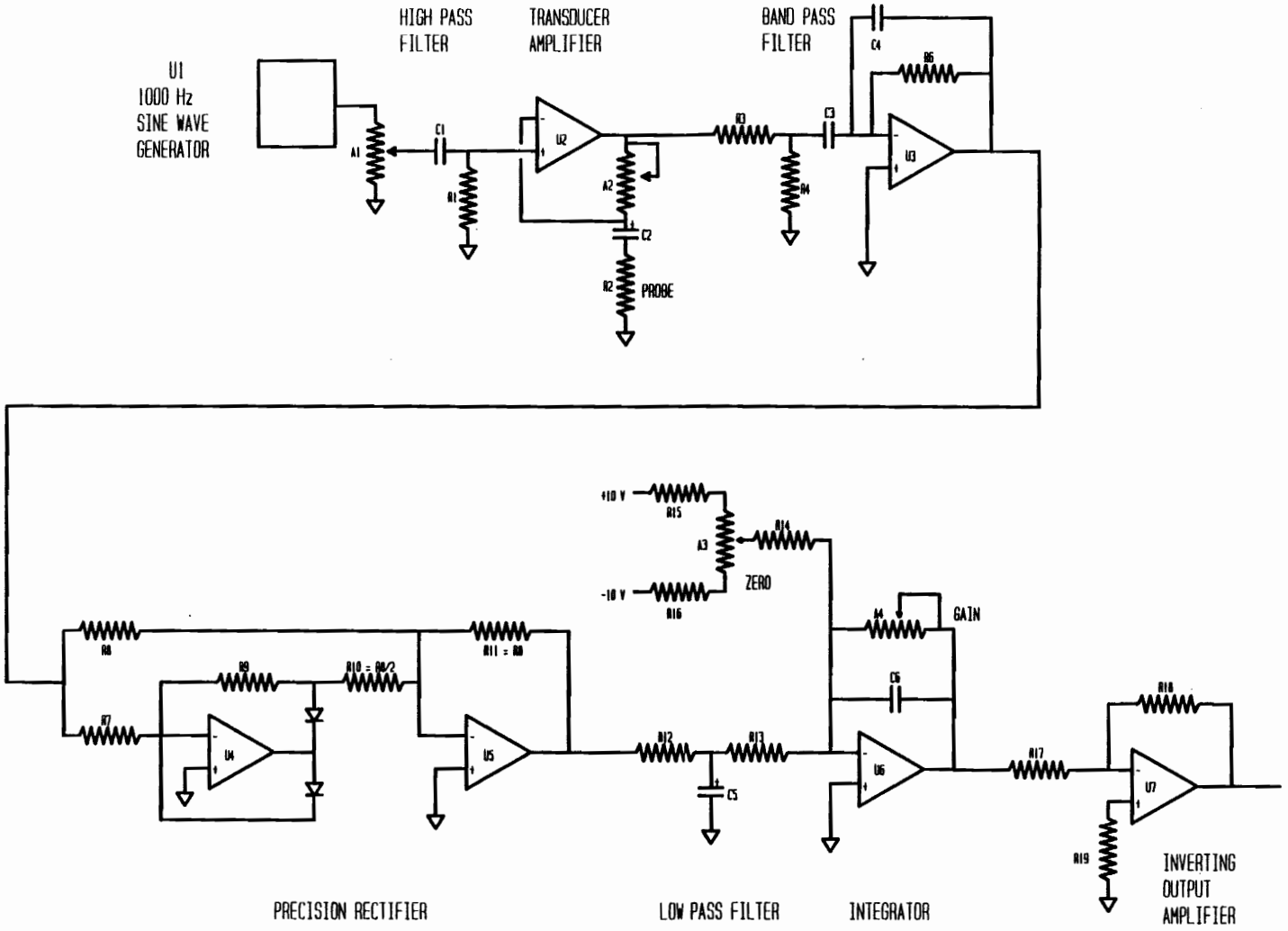


Figure A.5.1. Conductivity transducer circuit details.

TABLE A.5.1

CIRCUIT COMPONENT VALUES

Potentiometers K ohms		Resistors K ohms		Capacitors μF		I.C.'s	
A1	100	R1	15	C1	0.1	U1	INTERSIL 8038
A2	100	R2	probe	C2	1.0	U2	LM324N
A3	1	R3	22	C3	0.01	U3	LM324N
A4	100	R4	5.6	C4	0.01	U4	LM324N
		R6	47	C5	10.0	U5	LM324N
		R7	10	C6	0.25	U6	LM324N
		R8	10			U7	LM324N
		R9	10				
		R10	5				
		R11	10				
		R12	5				
		R13	5				
		R14	10				
		R15	2				
		R16	2				
		R17	10				
		R18	10				
		R19	10				

APPENDIX 6

COMPUTER CODE LISTINGS

FORTRAN PROGRAM 'STAGES': Optimizes the number of theoretical stages in the staged cascade model with respect to a given experimental data set. The optimization is carried out in the frequency domain using an accelerated gradient search. The data set has been transformed into the frequency domain by a FFT code (not included). The objective function, i.e. the model impulse response equation, is written in the frequency domain as its Laplace Transform.

FORTRAN PROGRAM 'DTIME': Duplicates program 'STAGES' for the staged cascade model modified to include a dead time.

FORTRAN PROGRAM 'YTCALC': Numerically evaluates the above ideal models' unit impulse response for direct comparison with the corresponding experimental data.

FORTRAN PROGRAM 'HEAT': The thin-film processor transient heat transfer simulation program. The calling program assigns values to the dimensional and thermal variables of the processor, and the process fluid. The IMSL subroutine DVERK integrates the staged cascade model as a linear system of equations. The IMSL subroutine DPDES integrates the thermal wall temperature as a function of position and time.

SAS PROGRAM 'COEFFICIENTS FOR A QUADRATIC HEAT TRANSFER SPACE MODEL': An adaptation of SAS 'PROC REG' to predict the location of a 'minimum' on a quadratic response surface of 'inside' and 'outside' heat transfer film coefficients.

SAS PROGRAM (un titled): Invokes SAS 'PROC MATRIX' to evaluate the 'inside' and 'outside' heat transfer film coefficients from the quadratic response coefficients determined in 'PROC REG' above.

```
C*************************************************************************
C
C
C  PROGRAM NAME:  STAGES
C
C
C*************************************************************************
C
C  WRITTEN BY: KIM HUNTER                        DATE: TRANSCRIBED
C                                                       5-20-86
C
C
C  SOURCE: "OPTIMIZATION TECHNIQUES WITH FORTRAN", KUESTER & MIZE,
C          MC GRAW-HILL, N. Y. (1973).
C
C          CODE MODIFIED TO "STAGES FORTRAN" STANDARD   DATE: 4-21-87
C
C  PURPOSE: THIS CODE LOCATES A LOCAL STATIONARY POINT (MAXIMUM,
C          MINIMUM, OR SADDLE POINT) OF AN OBJECTIVE FUNCTION ON
C          A MULTIDIMENSIONAL SURFACE.
C
C
C
C*************************************************************************
C
C
C     PARAMETERS:
C                 N = NUMBER OF THEORETICAL STAGES ASSIGNED TO MODEL.
C                 " = INDEX ON THE STAGE INCREMENTING LOOP.
C                 PASS1 = (LOGICAL) FLAG THAT PERMITS SEARCH PARAMETERS
C                         TO BE OUTPUT ONLY ON THE FIRST PASS OF THE
C                         STAGE INCREMENTING LOOP - FOR "IPRINT" = 0.
C
C
C                 NSTAGE = NUMBER OF INDEPENDENT VARIABLES.
C                 NPTS = NUMBER OF DATA POINTS READ.
C                 RK = VECTOR OF INITIAL GUESSES FOR THE
C                      INDEPENDENT VARIABLES.
C                 EPS = VECTOR OF INITIAL STEP SIZES FOR THE
C                       INDEPENDENT VARIABLES.
C                 ITMAX = LIMIT ON THE NUMBER OF TIMES THE
C                         OBJECTIVE FUNCTION ROUTINE IS CALLED.
C                 NKAT = NUMBER OF TIMES INITIAL STEP SIZE IS TO
C                        BE REDUCED.
C                 EPSY = MAXIMUM ERROR IN OBJECTIVE FUNCTION
C                        FOR CONVERGENCE.
C                 ALPHA = FACTOR FOR EXTENDING INITIAL STEP SIZE -
C                         MUST BE .GE. 1.
C                 BETA = FACTOR FOR REDUCING INITIAL STEP SIZE -
C                        MUST BE .GE. ZERO AND .LE. 1.
C                 QD = OPTIMUM VALUE OF OBJECTIVE FUNCTION RESULTING
C                      FROM THE SEARCH.
C                 IPRINT = PRINT CONTROL PARAMETER.
C                          IPRINT = 0;  NO INTERMEDIATE RESULTS.
C                          IPRINT = 1;  RESULTS AT EACH ITERATION.
C
C
C                 AKE = VECTOR OF INDEPENDENT VARIABLES IN
C                       SUBROUTINE OBJECT.
C                 SUMN = VALUE OF THE OBJECTIVE FUNCTION CALCULATED
C                        IN SUBROUTINE OBJECT.
C
C
C
C        THE HOOKE AND JEEVES PATTERN SEARCH USES TWO SUBROUTINES
C
C  1.  SUBROUTINE HOOKE CARRIES OUT THE FOLLOWING TASKS....
```

```
C          A) SET-UP AND EXECUTION OF THE SEARCH PATTERN.
C          B) ACCELERATION OF THE PATTERN BY INCREASING STEP SIZE.
C          C) TESTING FOR PATTERN FAILURE AND RESTARTING THE
C             SEARCH AT THE LAST VALID BASE POINT WHEN FAILURE IS
C             DETECTED.
C          E) TESTING FOR OVERALL CONVERGENCE OF THE OBJECTIVE FUNC.
C             AND REDUCING INITIAL STEP SIZE NEAR THE OPTIMUM POINT.
C
C  2.  SUBROUTINE OBJECT EVALUATES THE OBJECTIVE FUNCTION.  THIS
C      SUBPROGRAM MUST BE SUPPLIED BY THE USER
C
C
C**********************************************************************
C
C
C               CALLING PROGRAM FOR SUBROUTINE HOOKE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS
      LOGICAL PASS1
      DIMENSION EPS(10), RK(10), Q(10), QQ(10), W(10),
     &          X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
      NI = 5
      NO = 6
C
C  ENTER THE NO. OF PARAMETERS TO EVALUATE 'NSTAGE' AND NO. OF
C  DATA POINTS 'NPTS'.
C
      NSTAGE = 2
      NPTS = 20
      IPRINT = 0
      ITMAX = 500
      NKAT = 20
      EPSY = 1.D-4
      ALPHA = 1.D0
      BETA = .5D0
C
      QD = 0.0D0
C
C  READ IN INDEPENDENT AND DEPENDENT VARIABLES
C
      WRITE (NO,603)
C
      DO 601 I=1,NPTS
         READ (NI,602) X(I), YOBS(I)
         WRITE (NO,604) X(I), YOBS(I)
 601  CONTINUE
C
 600  FORMAT (I3)
 602  FORMAT (4X,3(F8.5,1X))
 603  FORMAT ('1',////,T15,'FREQUENCY RESPONSE INPUT DATA',
     &        //,T15,'FREQUENCY 1/S',14X,'COMPLEX RESPONSE (RE,IM)',//)
 604  FORMAT (1X,T15,D12.5,T41,'( ',D12.5,' , ',D12.5,' )')
C
C.....................................................................
C
C  THE FOLLOWING CODE LOOP CARRIES OUT THE PATTERN SEARCH FOR
C  SEQUENTIALLY VARYING VALUES OF "N" THE NUMBER OF STAGES.
C
C  THE AREA UNDER THE IMPULSE RESPONSE CURVE "AREA", AND THE
C  THE RESIDENCE TIME OF AN INDIVIDUAL STAGE "TAU" ARE OPTIMIZED
C  AT EACH SELECTED "N".....NOTE, THE STARTING GUESS FOR TAU
C  IS THE TOTAL SYSTEM RESIDENCE TIME "TAUTOT" DIVIDED BY THE NO. OF
C  STAGES N.
C
```

```
C  BEST FIT VALUES OF THE PARAMETERS AND THE RESPONSE ARE OUTPUT AT
C  THE END OF EACH PASS.
C
C
      PASS1 = .TRUE.
      NSTART = 10
      DO 30 N=NSTART, NSTART + 24
C
C        STARTING GUESSES
C
         TAUTOT = 2.0D0
         TAU = TAUTOT / DBLE(N)
         AREA = 19.99D0

         RK(1) = 0.1D0
         RK(2) = 19.99D0
C
C        STARTING STEP SIZES
C
         EPS(1) = .1D0
         EPS(2) = .1D0
C
      CALL HOOKE (RK, EPS, NSTAGE, ITMAX, NKAT, EPSY, ALPHA, BETA, QD,
     &           Q, QQ, W, IPRINT, PASS1)
C
      PASS1 = .FALSE.
 30   CONTINUE
      STOP
      END
C
C
C*********************************************************************
C
C                         SUBROUTINE HOOKE
C
C*********************************************************************
C
C
      SUBROUTINE HOOKE (RK, EPS, NSTAGE, MAXK, NKAT, EPSY, ALPHA,
     &                  BETA, QD, Q, QQ, W, IPRINT, PASS1)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS
      LOGICAL PASS1
      DIMENSION EPS(NSTAGE), RK(NSTAGE), Q(NSTAGE), QQ(NSTAGE),
     &          W(NSTAGE), X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
C  OUTPUT SEARCH ROUTINE PARAMETERS FOR FIRST PASS ONLY.
C
      IF (PASS1) THEN
C
      WRITE (NO,001)
 001  FORMAT (////,15X,'HOOKE AND JEEVES OPTIMIZATION ROUTINE')
      WRITE (NO,002) ALPHA, BETA, MAXK, NKAT
 002  FORMAT (//,15X,'PARAMETERS',/15X,'ALPHA = ',D14.7,4X,
     &        'BETA = ',D14.7,/15X,'ITMAX = ',I4,4X,'NKAT = ',I3)
      WRITE (NO,003) NSTAGE
 003  FORMAT (/,15X,'NUMBER OF INDEPENDENT VARIABLES = ',I3)
      WRITE (NO,004)
 004  FORMAT (/,15X,'INITIAL STEP SIZES')
      DO 6 I=1,NSTAGE
         WRITE (NO,005) I, EPS(I)
 005     FORMAT (/,15X,'EPS(',I2,') = ',D14.7)
 6    CONTINUE
      WRITE (NO,007) EPSY
 007  FORMAT (/,15X,'ERROR IN FUNCTION VALUES FOR CONVERGENCE = ',
```

```
     &        D14.7)
      WRITE (NO,1000)
 1000 FORMAT ('1',////,T15,'SUM OF SQUARES ERROR AS A FUNCTION',1X,
     &        'OF NO. OF STAGES')
      WRITE (NO,1001)
 1001 FORMAT (////,T15,'FEED RATE =      KG/HR (    LBM/HR)')
      WRITE (NO,1002)
 1002 FORMAT (/,T15,'ROTOR SPEED =      M/S (     RPM)')
      WRITE (NO,008)
 008  FORMAT (//,T15,'NO. STAGES',T27,'SSQ ERROR',T36,'    TAU  ',
     &        T45,'    AREA  ',/)
C
C
      ENDIF
C
C.....................................................................
C
C  THE FOLLOWING CODE SEGMENT INITIALIZES SUBROUTINE HOOKE PARAMS.
C
C           KAT = REDUCTIONS IN INITIAL STEP SIZE.
C           KK1 = TOTAL NUMBER OF FUNCTION EVALUATIONS.
C           KCOUNT = NUMBER OF FAILED AXES FROM A SEARCH POINT.
C           Q(I) = COORDINATES OF LAST SUCCESSFUL BASE POINT.
C           QQ(I) =     "      "  NEXT SEARCH POINT.
C           KFLAG = 1 IF PREVIOUS SEARCH POINT FAILED ON ALL AXES.
C           W(I) = WORK VECTOR FOR HOLDING OBJ. FUNC. VALUES.
C
C
      KFLAG = 0
      DO 601 I=1,NSTAGE
         Q(I) = RK(I)
         W(I) = 0.0D0
 601  CONTINUE
C
      KAT = 0
      KK1 = 0
 70   KCOUNT = 0
      WBEST = W(NSTAGE)
C
C  THE FOLLOWING SEGMENT TESTS THE CURRENT BASE POINT.  IF THIS
C  POINT IS NOT AN IMPROVEMENT IT IS TAGGED BY SETTING KFLAG = 1.
C
      CALL OBJECT (SUM, RK, NSTAGE)
C
      KK1 = KK1 + 1
      BO = SUM
      IF (KK1 .EQ. 1) QD = SUM
      IF (KK1 .EQ. 1) GO TO 201
      IF (BO .GT. QD) KFLAG = 1
      IF (BO .LT. QD) QD = BO
C
C.....................................................................
C
C  THE FOLLOWING SEGMENT ESTABLISH THE SEARCH PATTERN.  VECTOR QQ
C  IS THE STARTING POINT FOR THE NEXT STEP.  EACH PASS THRU THE
C  LOOP EXTENDS THE SEARCH IN ONE DIRECTION.  IF THE OBJ. FUNC. IS
C  IMPROVED THE SOLUTION VECTOR RK(I) IS UPDATED AND THE SEARCH
C  CONTINUED.  IF NOT THE STEP SIZE IS HALVED AND THE SEARCH REPEATED
C  IN THAT DIRECTION.  IF NO IMPROVEMENT ON THE SECOND TRY THE AXIS
C  IS SAID TO HAVE FAILED AND KCOUNT IS INCREMENTED.  THE SOLUTION
C  COORDINATE RK(I) AND OBJ. FUNC. W(I) ARE BACK DATED TO THEIR
C  PREVIOUS VALUES TSRK AND BO.
C
 201  DO 55 I=1,NSTAGE
         QQ(I) = RK(I)
         TSRK = RK(I)
```

```
          RK(I) = RK(I) + EPS(I)
C
          CALL OBJECT (SUM, RK, NSTAGE)
C
          KK1 = KK1 + 1
          W(I) = SUM
          IF (W(I) .LT. QD) GO TO 58
          RK(I) = RK(I) - 2.0D0 * EPS(I)
C
          CALL OBJECT (SUM, RK, NSTAGE)
C
          KK1 = KK1 + 1
          W(I) = SUM
          IF (W(I) .LT. QD) GO TO 58
          RK(I) = TSRK
          IF (I .EQ. 1) GO TO 513
          W(I) = W(I-1)
          GO TO 613
  513     W(I) = BO
  613     CONTINUE
          KCOUNT = KCOUNT + 1
          GO TO 55
 58       QD = W(I)
          QQ(I) = RK(I)
 55    CONTINUE
C
C.....................................................................
C
       IF (IPRINT) 60, 65, 60
 60    WRITE (NO,100) KK1
C
C  RECORD RESPONSES AND LOCATION
C
       WRITE (NO,102)
       WRITE (NO,207) (RK(I), I=1,NSTAGE), QD
C
C  TEST TO DETERMINE TERMINATION OF PROGRAM.
C
 65    IF (KK1 .GT. MAXK) GO TO 94
       IF (KAT .GE. NKAT) GO TO 94
       IF (DABS(W(NSTAGE)-WBEST) .LE. EPSY) GO TO 94
C
C.....................................................................
C
C  THE FOLLOWING CODE SEGMENT ADJUSTS SEARCH STEP SIZE
C
C  IF ALL AXES FAIL REDUCE STEP SIZE.
C  OTHERWISE, INCREASE STEP SIZE.
C
       IF (KCOUNT .GE. NSTAGE) GO TO 28
C
       DO 26 I=1,NSTAGE
          RK(I) = RK(I) + ALPHA * (RK(I) - Q(I))
 26    CONTINUE
C
       DO 25 I=1,NSTAGE
          Q(I) = QQ(I)
 25    CONTINUE
C
       GO TO 70
C
C  REDUCE STEP SIZE
C
 28    KAT = KAT + 1
C
C  IF THE CURRENT POINT DOES NOT IMPROVE THE OBJECTIVE FUNCTION
```

```
C  MAKE THE LAST BASE POINT THE NEW STARTING POINT.
C
      IF (KFLAG .EQ. 1) GO TO 202
      GO TO 204
 202  KFLAG = 0
C
      DO 203 I=1,NSTAGE
         RK(I) = Q(I)
 203  CONTINUE
C
 204  DO 80 I=1,NSTAGE
         EPS(I) = EPS(I) * BETA
 80   CONTINUE
C
C.....................................................................
C
      IF (IPRINT) 85, 70, 85
 85   WRITE (NO,101) KAT
      GO TO 70
C
 94   IF (IPRINT .NE. 0) THEN
C
      IF (IPRINT) 96, 95, 96
 96   WRITE (NO,460) (EPS(I), I=1,NSTAGE)
      WRITE (NO,461) (RK(I), I=1,NSTAGE)
 95   WRITE (NO,462) QD
C
      DO 104 I=1,NSTAGE
         WRITE (NO,103) I, RK(I)
 104  CONTINUE
C
      WRITE (NO,100) KK1
C
      ENDIF
C
      WRITE (NO,463) N, QD, (RK(I),I=1,NSTAGE)
C
C
 100  FORMAT (//,2X,'NUMBER OF FUNCTION EVALUATIONS = ',I8)
 101  FORMAT (/,2X,'STEP SIZE REDUCED ',I2,' TIMES')
 102  FORMAT (1X,'END OF EACH PATTERN SEARCH',/)
 103  FORMAT (//,2X,'FINAL X(',I2,') = ',D14.7)
 207  FORMAT (1X,'VARIABLES AND SUMN',3X,9D12.4//)
 465  FORMAT (10X,'SUM',3X,D14.7)
 460  FORMAT (1X,'THE FINAL EPS ARE,   ',4D14.7/)
 461  FORMAT (1X,'THE FINAL RK ARE,    ',5D14.7/)
 462  FORMAT (1X,'THE MINIMUM RESPONSE IS,   ',D14.7/)
 463  FORMAT (20X,I3,2X,E10.3,3(1X,F8.4,1X))
C
      RETURN
      END
C
C
C
C
C**********************************************************************
C
C                     SUBROUTINE OBJECT
C
C  THE OBJECTIVE FUNCTION EVALUATED BY THIS SUBROUTINE IS THE
C  FREQUENCY RESPONSE OF AN N-TH ORDER SYSTEM WHOSE TRANSFER FUNCTION
C  HAS THE FORM:    Y(S) = AREA * EXP(-D*S) * [A/( S + A )] **N
C
C         WHERE:   S = LAPLACE VARIABLE  J * (2*PI*FREQ.)
C                  AREA = THE AREA UNDER THE RESIDENCE TIME DISTRIBUTION
C                         CURVE IN UNITS OF (RESPONSE * SAMPLING RATE).
C                         THIS REPRESENTS THE STEADY STATE D.C. SIGNAL
```

```
C                          LEVEL OF THE TRANSFORMED RESPONSE IN THE
C                          FREQUENCY DOMAIN.  NOTE, THE EQUIVALENT AREA
C                          UNDER THE RTD CURVE CAN BE OBTAINED DIRECTLY
C                          BY DIVIDING "AREA" BY THE SAMPLING RATE OR,
C                          MULTIPLYING BY THE SAMPLE INTERVAL.
C
C
C                    D = THE CHARACTERISTIC "DEAD TIME" THAT PRECEDES
C                        THE SYSTEM RESPONSE.
C
C                   TAU = THE CHARACTERISTIC TIME CONSTANT (MEAN RESIDENCE
C                         TIME) OF A SINGLE STAGE.
C                   A = 1/TAU
C
C                   RN = THE NUMBER OF STAGES IN THE SYSTEM
C
C
C
C**********************************************************************
C
C
C
      SUBROUTINE OBJECT (SUMN, AKE, NSTAGE)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS, YCALC, TERMA, S
      DIMENSION AKE(NSTAGE), X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
      TAU1 = AKE(1)
      AREA = AKE(2)
      D = 0.D0
C
C  TO PREVENT POSSIBLE DIVISION BY ZERO.
C
      IF (TAU1 .LT. 1.D-6) TAU1 = 1.D-6
C
      A = 1.D0/TAU1
C
      SUMN = 0.D0
      DO 10 I=1,NPTS
         S = DCMPLX (0.D0, X(I)*2.D0*3.14159D0)
         TERMA = DCMPLX (A, X(I)*2.D0 *3.14159D0)
         YCALC = AREA* (A/TERMA)** N
         YCALC = CDEXP (-D*S) * YCALC
C     WRITE (NO,1) NPTS, YOBS(I), YCALC, SUMN
C1    FORMAT (1X,I3,1X,2(D10.3,1X))
C
C  THIS IS THE COMPLEX ERROR ESTIMATOR RECOMMENDED BY ANDERSEN & WHITE
C
         SUMN = SUMN + (CDABS( YOBS(I) - YCALC ))**2
 10   CONTINUE
C
      RETURN
      END
```

```
C*******************************************************************************
C
C
C  PROGRAM NAME:  DTIME
C
C
C*******************************************************************************
C
C  WRITTEN BY: A. I. JOHNSON                    DATE: TRANSCRIBED
C              UNIVERSITY OF                          5-20-86
C              WESTERN ONTARIO
C
C  SOURCE: "OPTIMIZATION TECHNIQUES WITH FORTRAN", KUESTER & MIZE,
C          MC GRAW-HILL, N. Y. (1973).
C
C          CODE MODIFIED TO "DTIME FORTRAN" PROGRAM   DATE: 4-21-87
C
C
C  PURPOSE: THIS CODE LOCATES A LOCAL STATIONARY POINT (MAXIMUM,
C          MINIMUM, OR SADDLE POINT) OF AN OBJECTIVE FUNCTION ON
C          A MULTIDIMENSIONAL SURFACE.
C
C
C
C*******************************************************************************
C
C
C     PARAMETERS:
C                N = NUMBER OF THEORETICAL STAGES ASSIGNED TO MODEL.
C                " = INDEX ON THE STAGE INCREMENTING LOOP.
C                NSTART = STARTING POINT FOR SEARCH
C                PASS1 = (LOGICAL) FLAG THAT PERMITS SEARCH PARAMETERS
C                        TO BE OUTPUT ONLY ON THE FIRST PASS OF THE
C                        STAGE INCREMENTING LOOP - FOR "IPRINT" = 0.
C
C
C                NSTAGE = NUMBER OF INDEPENDENT VARIABLES.
C                NPTS = NUMBER OF DATA POINTS READ.
C                RK = VECTOR OF INITIAL GUESSES FOR THE
C                     INDEPENDENT VARIABLES.
C                EPS = VECTOR OF INITIAL STEP SIZES FOR THE
C                      INDEPENDENT VARIABLES.
C                ITMAX = LIMIT ON THE NUMBER OF TIMES THE
C                        OBJECTIVE FUNCTION ROUTINE IS CALLED.
C                NKAT = NUMBER OF TIMES INITIAL STEP SIZE IS TO
C                       BE REDUCED.
C                EPSY = MAXIMUM ERROR IN OBJECTIVE FUNCTION
C                       FOR CONVERGENCE.
C                ALPHA = FACTOR FOR EXTENDING INITIAL STEP SIZE -
C                        MUST BE .GE. 1.
C                BETA = FACTOR FOR REDUCING INITIAL STEP SIZE -
C                       MUST BE .GE. ZERO AND .LE. 1.
C                QD = OPTIMUM VALUE OF OBJECTIVE FUNCTION RESULTING
C                     FROM THE SEARCH.
C                IPRINT = PRINT CONTROL PARAMETER.
C                         IPRINT = 0;  NO INTERMEDIATE RESULTS.
C                         IPRINT = 1;  RESULTS AT EACH ITERATION.
C
C
C                AKE = VECTOR OF INDEPENDENT VARIABLES IN
C                      SUBROUTINE OBJECT.
C                SUMN = VALUE OF THE OBJECTIVE FUNCTION CALCULATED
C                       IN SUBROUTINE OBJECT.
C
C
C
C        THE HOOKE AND JEEVES PATTERN SEARCH USES TWO SUBROUTINES
```

```
C
C  1.  SUBROUTINE HOOKE CARRIES OUT THE FOLLOWING TASKS....
C          A) SET-UP AND EXECUTION OF THE SEARCH PATTERN.
C          B) ACCELERATION OF THE PATTERN BY INCREASING STEP SIZE.
C          C) TESTING FOR PATTERN FAILURE AND RESTARTING THE
C             SEARCH AT THE LAST VALID BASE POINT WHEN FAILURE IS
C             DETECTED.
C          E) TESTING FOR OVERALL CONVERGENCE OF THE OBJECTIVE FUNC.
C             AND REDUCING INITIAL STEP SIZE NEAR THE OPTIMUM POINT.
C
C  2.  SUBROUTINE OBJECT EVALUATES THE OBJECTIVE FUNCTION.  THIS
C      SUBPROGRAM MUST BE SUPPLIED BY THE USER
C
C
C*********************************************************************
C
C
C               CALLING PROGRAM FOR SUBROUTINE HOOKE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS
      LOGICAL PASS1
      DIMENSION EPS(10), RK(10), Q(10), QQ(10), W(10),
     &          X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
      NI = 5
      NO = 6
C
C  ENTER THE NO. OF PARAMETERS TO EVALUATE 'NSTAGE' AND NO. OF
C  DATA POINTS 'NPTS'.
C
      NSTAGE = 3
      NPTS = 20
      IPRINT = 0
      ITMAX = 500
      NKAT = 20
      EPSY = 1.D-4
      ALPHA = 1.D0
      BETA = .5D0
C
      QD = 0.0D0
C
C  READ IN INDEPENDENT AND DEPENDENT VARIABLES
C
C     WRITE (NO,603)
C
      DO 601 I=1,NPTS
         READ (NI,602) X(I), YOBS(I)
C        WRITE (NO,604) X(I), YOBS(I)
 601  CONTINUE
C
 600  FORMAT (I3)
 602  FORMAT (4X,3(F8.5,1X))
 603  FORMAT ('1',T10,'FREQUENCY 1/S',14X,'COMPLEX RESPONSE (RE,IM)',//)
 604  FORMAT (1X,T10,D12.5,T36,'( ',D12.5,' , ',D12.5,' )')
C
C.....................................................................
C
C  THE FOLLOWING CODE LOOP CARRIES OUT THE PATTERN SEARCH FOR
C  THE OPTIMUM SYSTEM DEAD TIME "D".
C
C  VALUES OF "N" THE NUMBER OF STAGES ARE VARIED ABOUT A
C  PREVIOUSLY DETERMINED BEST GUESS VALUE N = NSTART.  THE DEAD
C  TIME PARAMETER "D" AND THE SINGLE STAGE TIME CONSTANT "TAU" ARE
C  OPTIMIZED ABOUT THIS VALUE OF "N".
C
```

```
C  BEST FIT VALUES OF THE PARAMETERS AND THE RESPONSE ARE OUTPUT AT
C  THE END OF EACH PASS.
C
C
      PASS1 = .TRUE.
      NSTART = 1
      DO 30 N=NSTART  , NSTART+11
C
C        STARTING GUESSES
C
         AREA = 19.9D0
         TAU = .2D0
         D = 1.0D0
C
         RK(1) = TAU
         RK(2) = AREA
         RK(3) = D
C
C        STARTING STEP SIZES
C
         EPS(1) = .01D0
         EPS(2) = .1D0
         EPS(3) = .1D0
C
      CALL HOOKE (RK, EPS, NSTAGE, ITMAX, NKAT, EPSY, ALPHA, BETA, QD,
     &           Q, QQ, W, IPRINT, PASS1)
C
      PASS1 = .FALSE.
 30   CONTINUE
      STOP
      END
C
C
C*********************************************************************
C
C                          SUBROUTINE HOOKE
C
C*********************************************************************
C
C
      SUBROUTINE HOOKE (RK, EPS, NSTAGE, MAXK, NKAT, EPSY, ALPHA,
     &                  BETA, QD, Q, QQ, W, IPRINT, PASS1)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS
      LOGICAL PASS1, PAROUT
      DIMENSION EPS(NSTAGE), RK(NSTAGE), Q(NSTAGE), QQ(NSTAGE),
     &          W(NSTAGE), X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
C  PAROUT = .FALSE. SUPPRESSES OUTPUT OF ALL HOOKE AND JEEVES PARAMETERS
C
C  OUTPUT SEARCH ROUTINE PARAMETERS FOR FIRST PASS ONLY.
C
      IF (PASS1) THEN
C
      PAROUT = .FALSE.
      IF (.NOT. PAROUT) GO TO 1003
C
      WRITE (NO,001)
 001  FORMAT ('1',10X,'HOOKE AND JEEVES OPTIMIZATION ROUTINE')
      WRITE (NO,002) ALPHA, BETA, MAXK, NKAT
 002  FORMAT (//,2X,'PARAMETERS',/2X,'ALPHA = ',D14.7,4X,
     &        'BETA = ',D14.7,4X,'ITMAX = ',I4,4X,'NKAT = ',I3)
      WRITE (NO,003) NSTAGE
 003  FORMAT (/,2X,'NUMBER OF INDEPENDENT VARIABLES = ',I3)
      WRITE (NO,004)
```

```
 004  FORMAT (/,2X,'INITIAL STEP SIZES')
      DO 6 I=1,NSTAGE
         WRITE (NO,005) I, EPS(I)
 005     FORMAT (/,2X,'EPS(',I2,') = ',D14.7)
 6    CONTINUE
      WRITE (NO,007) EPSY
 007  FORMAT (/,2X,'ERROR IN FUNCTION VALUES FOR CONVERGENCE = ',
     &        D14.7)
C
 1003 CONTINUE
C
      WRITE (NO,1000)
 1000 FORMAT (////,T12,'SUM OF SQUARES ERROR AS A FUNCTION',1X,
     &        'OF NO. OF STAGES & DEAD TIME')
C     WRITE (NO,1001)
 1001 FORMAT (////,T15,'FEED RATE =     KG/HR (    LBM/HR)')
C     WRITE (NO,1002)
 1002 FORMAT (/,T15,'ROTOR SPEED =      M/S (     RPM)')
      WRITE (NO,008)
 008  FORMAT (//,T15,'NO. STAGES',T27,'SSQ ERROR',T36,'    TAU  ',
     &        T45,'    AREA  ',T58,'DEAD TIME',/)
C
C
      ENDIF
C
C.....................................................................
C
C  THE FOLLOWING CODE SEGMENT INITIALIZES SUBROUTINE HOOKE PARAMS.
C
C           KAT = REDUCTIONS IN INITIAL STEP SIZE.
C           KK1 = TOTAL NUMBER OF FUNCTION EVALUATIONS.
C           KCOUNT = NUMBER OF FAILED AXES FROM A SEARCH POINT.
C           Q(I) = COORDINATES OF LAST SUCCESSFUL BASE POINT.
C           QQ(I) =     "      "  NEXT SEARCH POINT.
C           KFLAG = 1 IF PREVIOUS SEARCH POINT FAILED ON ALL AXES.
C           W(I) = WORK VECTOR FOR HOLDING OBJ. FUNC. VALUES.
C
C
      KFLAG = 0
      DO 601 I=1,NSTAGE
         Q(I) = RK(I)
         W(I) = 0.0D0
 601  CONTINUE
C
      KAT = 0
      KK1 = 0
 70   KCOUNT = 0
      WBEST = W(NSTAGE)
C
C  THE FOLLOWING SEGMENT TESTS THE CURRENT BASE POINT.  IF THIS
C  POINT IS NOT AN IMPROVEMENT IT IS TAGGED BY SETTING KFLAG = 1.
C
      CALL OBJECT (SUM, RK, NSTAGE)
C
      KK1 = KK1 + 1
      BO = SUM
      IF (KK1 .EQ. 1) QD = SUM
      IF (KK1 .EQ. 1) GO TO 201
      IF (BO .GT. QD) KFLAG = 1
      IF (BO .LT. QD) QD = BO
C
C.....................................................................
C
C  THE FOLLOWING SEGMENT ESTABLISH THE SEARCH PATTERN.  VECTOR QQ
C  IS THE STARTING POINT FOR THE NEXT STEP.  EACH PASS THRU THE
C  LOOP EXTENDS THE SEARCH IN ONE DIRECTION.  IF THE OBJ. FUNC. IS
C  IMPROVED THE SOLUTION VECTOR RK(I) IS UPDATED AND THE SEARCH
```

```
C  CONTINUED.  IF NOT THE STEP SIZE IS HALVED AND THE SEARCH REPEATED
C  IN THAT DIRECTION.  IF NO IMPROVEMENT ON THE SECOND TRY THE AXIS
C  IS SAID TO HAVE FAILED AND KCOUNT IS INCREMENTED.  THE SOLUTION
C  COORDINATE RK(I) AND OBJ. FUNC. W(I) ARE BACK DATED TO THEIR
C  PREVIOUS VALUES TSRK AND BO.
C
 201   DO 55 I=1,NSTAGE
          QQ(I) = RK(I)
          TSRK = RK(I)
          RK(I) = RK(I) + EPS(I)
C
          CALL OBJECT (SUM, RK, NSTAGE)
C
          KK1 = KK1 + 1
          W(I) = SUM
          IF (W(I) .LT. QD) GO TO 58
          RK(I) = RK(I) - 2.0D0 * EPS(I)
C
          CALL OBJECT (SUM, RK, NSTAGE)
C
          KK1 = KK1 + 1
          W(I) = SUM
          IF (W(I) .LT. QD) GO TO 58
          RK(I) = TSRK
          IF (I .EQ. 1) GO TO 513
          W(I) = W(I-1)
          GO TO 613
   513    W(I) = BO
   613    CONTINUE
          KCOUNT = KCOUNT + 1
          GO TO 55
 58       QD = W(I)
          QQ(I) = RK(I)
 55    CONTINUE
C
C.....................................................................
C
       IF (IPRINT) 60, 65, 60
 60    WRITE (NO,100) KK1
C
C  RECORD RESPONSES AND LOCATION
C
       WRITE (NO,102)
       WRITE (NO,207) (RK(I), I=1,NSTAGE), QD
C
C  TEST TO DETERMINE TERMINATION OF PROGRAM.
C
 65    IF (KK1 .GT. MAXK) GO TO 94
       IF (KAT .GE. NKAT) GO TO 94
       IF (DABS(W(NSTAGE)-WBEST) .LE. EPSY) GO TO 94
C
C.....................................................................
C
C  THE FOLLOWING CODE SEGMENT ADJUSTS SEARCH STEP SIZE
C
C  IF ALL AXES FAIL REDUCE STEP SIZE.
C  OTHERWISE, INCREASE STEP SIZE.
C
       IF (KCOUNT .GE. NSTAGE) GO TO 28
C
       DO 26 I=1,NSTAGE
          RK(I) = RK(I) + ALPHA * (RK(I) - Q(I))
 26    CONTINUE
C
       DO 25 I=1,NSTAGE
          Q(I) = QQ(I)
```

```
 25   CONTINUE
C
      GO TO 70
C
C  REDUCE STEP SIZE
C
 28   KAT = KAT + 1
C
C  IF THE CURRENT POINT DOES NOT IMPROVE THE OBJECTIVE FUNCTION
C  MAKE THE LAST BASE POINT THE NEW STARTING POINT.
C
      IF (KFLAG .EQ. 1) GO TO 202
      GO TO 204
 202  KFLAG = 0
C
      DO 203 I=1,NSTAGE
         RK(I) = Q(I)
 203  CONTINUE
C
 204  DO 80 I=1,NSTAGE
         EPS(I) = EPS(I) * BETA
 80   CONTINUE
C
C.....................................................................
C
      IF (IPRINT) 85, 70, 85
 85   WRITE (NO,101) KAT
      GO TO 70
C
 94   IF (IPRINT .NE. 0) THEN
C
      IF (IPRINT) 96, 95, 96
 96   WRITE (NO,460) (EPS(I), I=1,NSTAGE)
      WRITE (NO,461) (RK(I), I=1,NSTAGE)
 95   WRITE (NO,462) QD
C
      DO 104 I=1,NSTAGE
         WRITE (NO,103) I, RK(I)
 104  CONTINUE
C
      WRITE (NO,100) KK1
C
      ENDIF
C
      WRITE (NO,463) N, QD, (RK(I),I=1,NSTAGE)
C
C
 100  FORMAT (//,2X,'NUMBER OF FUNCTION EVALUATIONS = ',I8)
 101  FORMAT (/,2X,'STEP SIZE REDUCED ',I2,' TIMES')
 102  FORMAT (1X,'END OF EACH PATTERN SEARCH',/)
 103  FORMAT (//,2X,'FINAL X(',I2,') = ',D14.7)
 207  FORMAT (1X,'VARIABLES AND SUMN',3X,9D12.4//)
 465  FORMAT (10X,'SUM',3X,D14.7)
 460  FORMAT (1X,'THE FINAL EPS ARE,   ',4D14.7/)
 461  FORMAT (1X,'THE FINAL RK ARE,    ',5D14.7/)
 462  FORMAT (1X,'THE MINIMUM RESPONSE IS,   ',D14.7/)
 463  FORMAT (20X,I3,2X,E10.3,3(1X,F8.4,1X))
C
      RETURN
      END
C
C
C
C
C**********************************************************************
C
C                     SUBROUTINE OBJECT
```

```
C
C  THE OBJECTIVE FUNCTION EVALUATED BY THIS SUBROUTINE IS THE
C  FREQUENCY RESPONSE OF AN N-TH ORDER SYSTEM WHOSE TRANSFER FUNCTION
C  HAS THE FORM:    Y(S) = AREA * EXP(-D*S) * [A/( S + A )] **N
C
C         WHERE:   S = LAPLACE VARIABLE  J * (2*PI*FREQ.)
C                  AREA = THE AREA UNDER THE RESIDENCE TIME DISTRIBUTION
C                         CURVE IN UNITS OF (RESPONSE * SAMPLING RATE).
C                         THIS REPRESENTS THE STEADY STATE D.C. SIGNAL
C                         LEVEL OF THE TRANSFORMED RESPONSE IN THE
C                         FREQUENCY DOMAIN.  NOTE, THE EQUIVALENT AREA
C                         UNDER THE RTD CURVE CAN BE OBTAINED DIRECTLY
C                         BY DIVIDING "AREA" BY THE SAMPLING RATE OR,
C                         MULTIPLYING BY THE SAMPLE INTERVAL.
C
C
C                   D = THE CHARACTERISTIC "DEAD TIME" THAT PRECEDES
C                       THE SYSTEM RESPONSE.
C
C                 TAU = THE CHARACTERISTIC TIME CONSTANT (MEAN RESIDENCE
C                       TIME) OF A SINGLE STAGE.
C                 A = 1/TAU
C
C                 RN = THE NUMBER OF STAGES IN THE SYSTEM
C
C
C
C**********************************************************************
C
C
C
      SUBROUTINE OBJECT (SUMN, AKE, NSTAGE)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 YOBS, YCALC, TERMA, S
      DIMENSION AKE(NSTAGE), X(300), YOBS(300)
      COMMON YOBS, X, NI, NO, NPTS, N
C
      TAU1 = AKE(1)
      AREA = AKE(2)
      D = AKE(3)
C
C  TO PREVENT POSSIBLE DIVISION BY ZERO.
C
      IF (TAU1 .LT. 1.D-6) TAU1 = 1.D-6
C
      A = 1.D0/TAU1
C
      SUMN = 0.D0
      DO 10 I=1,NPTS
         S = DCMPLX (0.D0, X(I)*2.D0*3.14159D0)
         TERMA = DCMPLX (A, X(I)*2.D0 *3.14159D0)
         YCALC = AREA* (A/TERMA)** N
         YCALC = CDEXP (-D*S) * YCALC
C     WRITE (NO,1) NPTS, YOBS(I), YCALC, SUMN
C1    FORMAT (1X,I3,1X,2(D10.3,1X))
C
C  THIS IS THE COMPLEX ERROR ESTIMATOR RECOMMENDED BY ANDERSEN & WHITE
C
         SUMN = SUMN + (CDABS( YOBS(I) - YCALC ))**2
 10   CONTINUE
C
      RETURN
      END
```

```
C*************************************************************************
C
C
C  PROGRAM NAME: YTCALC                    DATE WRITTEN: 3/87
C
C  WRITTEN BY: KIM HUNTER                  DATE REVISED: 6/7/87
C
C                                          REVISION NO.: 1
C
C.........................................................................
C
C
C  PURPOSE: TO NUMERICALLY EVALUATE THE IDEAL MODEL RESPONSE FUNCTION -
C
C    YTCALC = H(T-DT) * (1/TAU)**N * T**((N-1)/N!) * EXP((T-DT)/TAU)
C
C  WHERE:
C         YTCALC = MODEL NORMALIZED (UNIT) IMPULSE RESPONSE.
C         T = TIME....SEC.
C         TINCR = TIME INCREMENT....SEC.
C         N = NO. OF STAGES IN THE MODEL.
C         TAU = MEAN RESIDENCE TIME OF A SINGLE STAGE....SEC.
C         DEADT = MODEL SYSTEM "DEAD TIME"....SEC.
C         H() = THE TRANSLATED (DISPLACED) HEAVYSIDE FUNCTION
C               (UNIT STEP FUNCTION).
C
C
C*************************************************************************
C
C
C  FUNCTION SUBPROGRAM TO CALCULATE FACTORIAL N!
C
      FUNCTION FACT (N)
      FACT = 1.
      IF (N .LE. 1) RETURN
      DO 10 I=2,N
 10      FACT = FACT*REAL(I)
      RETURN
      END
C
      REAL NM1FAC
C
C.........................................................................
C
      N = 4
      TAU = .2766
      DEADT = 1.0812
C
      TINCR = 0.05
      KCOUNT = IFIX (10./TINCR)
C
      NM1 = N - 1
      FNM1 = FACT (NM1)
      T = 0.
      YTCALC = 0.
C
      DO 10 K=1,KCOUNT
         TIME = T-DEADT
C
C  APPLY THE HEAVYSIDE TRANSLATION FUNCTION...F(T-DT) = H(T-DT)*F(T).
C
         IF (TIME .LT. 0.) THEN
            YTCALC = 0.
         ELSE
C
C  EVALUATE THE IDEAL RESPONSE EQUATION.
C
```

```
            YTCALC = (1./TAU) ** N * TIME ** (N-1)/FNM1 *
     &       EXP( -1./TAU * TIME )
         ENDIF
         WRITE (6,100) T, YTCALC
         T = T+TINCR
 10    CONTINUE
       STOP
C
C  OUTPUT FORMATS.............................
C
 100   FORMAT (2(E14.7,2X))
       END
```

```
C*************************************************************************
C
C
C  PROGRAM NAME: HEAT
C
C
C*************************************************************************
C
C  WRITTEN BY: KIM HUNTER                    DATE: 6/25/86
C
C  REVISED BY: KIM HUNTER                    DATE: 7/18/87
C
C
C  PURPOSE: THIS PROGRAM USES A TANKS IN SERIES MODEL WITH LUMPED
C           PARAMETERS TO SIMULATE THE TRANSIENT PERFORMANCE OF A
C           THIN-FILM EVAPORATOR.
C
C           EACH STAGE OF THE CASCADE IS MODELED WITH TWO INTERACTIVE
C           ENERGY BALANCES.  ONE ON THE METAL HEAT TRANSFER WALL, AND
C           ONE ON THE FLUID IN THE TANK.  THE OVERALL ENERGY BALANCES
C           ARE:
C
C
C
C           ENERGY CONVECTED INTO THE STAGE BY FLUID FLOW - ENERGY
C           CONVECTED OUT BY FLUID FLOW + ENERGY CONDUCTED IN THROUGH
C           THE METAL WALL = RATE OF ACCUMULATION IN THE FLUID.
C
C
C           ENERGY CONVECTED INTO THE WALL ON THE HEATING SIDE - ENERGY
C           CONVECTED OUT OF THE WALL ON THE PROCESS SIDE = RATE OF
C           ENERGY ACCUMULATION IN THE WALL (RATE OF CHANGE OF WALL
C           TEMP)
C
C
C           WMASS * CPM * D(TEMP)/DT = HO * AREA * (TO - TWALL)      [1]
C
C
C
C           FMASS * CPF * D(TEMP)/DT = WF * CPF * (T - TEMPIN) +
C
C                                      HI * AREA (TWALL - T)  +      [2]
C
C                                      ROTOR POWER INPUT
C
C
C     THE ABOVE BALANCES ARE BASED ON A LUMPED PARAMETER MODEL OF THE
C     FLUID AND WALL ELEMENTS.  A MORE FAITHFUL SIMULATION OF THE
C     SYSTEM'S TIME RESPONSE CAN BE OBTAINED USING A DISTRIBUTED
C     PARAMETER MODEL FOR THE HEAT TRANSFER WALL, REPLACING EQU. [2]
C     WITH:
C
C
C     THE THERMAL 'DIFFUSION EQUATION' -
C
C          D2(TWALL)/DXDX = (TWALL*CPM / K*L) * DT/DTIME          [3]
C
C     BOUNDARY CONDITIONS -
C
C     AT X=0;  HO*AREA * (TO-TWALL) + K * AREA * D(TWALL)/DX = 0
C                                                                  [4]
C     AT X=L;  HI*AREA * (TWALL-T) + K * AREA * D(TWALL)/DX = 0
C
C
C
C
C     THE PROGRAM SOLVES THE SYSTEM OF SIMULTANEOUS O.D.E.S FOR
```

```
C     FLUID TEMPERATURES IN A THIN-FILM EVAPORATOR WHICH IS MODELED AS
C     A CASCADE OF N STAGES.  THE FLUID AND WALL TEMPERATURES ARE
C     CALCULATED AS FUNCTIONS OF TIME FOR EACH STAGE.  THE
C     FLUID TEMPERATURE RESPONSE IS FORMULATED AS AN INITIAL VALUE
C     PROBLEM AND IS INTEGRATED USING A SIXTH ORDER RUNGE-KUTTA
C     ROUTINE 'DVERK' SUPPLIED BY THE IMSL LIBRARY.  THE WALL TEMP-
C     ERATURE IS CALCULATED AS A BOUNDARY VALUE PROBLEM (NESTED INSIDE
C     THE I.V.P. SOLVER).  THE INSIDE WALL TEMPERATURE IS CALCULATED
C     FOR EACH STAGE USING A PARABOLIC P.D.E. SOLVER 'DPDES', ALSO
C     SUPPLIED BY THE IMSL LIBRARY.  'DPDES' PERFORMS A SPACE
C     DESCRITIZATION USING SPLINE COLLOCATION WITH A HERMITE CUBIC
C     BASIS.  THE TIME INTEGRATION IS CARRIED OUT BY METHOD OF LINES.
C
C
C
C.....................................................................
C
C
C                    MAIN PROGRAM VARIABLE DEFINITIONS:
C
C     TIME = INDEPENDENT VARIABLE OF INTEGRATION....SEC.
C     DTIME = TIME INTEGRATION STEP....SEC.
C     TEMP(N) & TNOW(N) = ARRAY CONTAINING FLUID TEMPERATURES...DEG C.
C     DTEMP(N) =                    "    DERIVATIVES  "       "
C     TLAST(N) = ARRAY CONTAINING FLUID TEMPS FROM LAST INTEGRATION
C                STEP....DEG C.
C     N = NUMBER OF STAGES IN THE CASCADE (N=L/2).
C     NSTAGE = LOOP COUNTER FOR SUBROUTINE DPDES.
C     AN = FLOATING POINT VALUE OF N
C     FMASS(N) = STEADY STATE MASS OF FLUID AT EACH STAGE....KG.
C     HOLDUP = TOTAL FLUID MASS IN UNIT (SUM OF FMASS(I)).
C     WMASS(N) = MASS OF WALL ELEMENT CORRESPONDING TO EACH STAGE.
C     AREA(N) =  AREA "   "      "          "          "     " ...SQ M.
C     TMWALL = TOTAL MASS OF HEAT TRANSFER WALL (SUM OF WMASS(I)).
C     TAREA = TOTAL HEAT TRANSFER AREA (SUM OF AREA(I)).
C     FLUX = LOGICAL VARIABLE TO FLAG FLUX OR NO FLUX INITIAL COND.
C     HO(N) = OUTSIDE (HEATING) WALL HEAT TRANSFER COEFFICIENT...
C               WATTS/SQ M-DEG K.
C     HI(N) = INSIDE (PROCESS) WALL H. T. COEFFICIENT.
C     AVHO = AVERAGED OUTSIDE H. T. COEFFICIENT.
C     AVHI = AVERAGED INSIDE H. T. COEFFICIENT.
C     TEMPIN = FLUID INLET TEMP TO THE CASCADE....DEG C.
C     TMPOUT = INITIAL FLUID OUTLET TEMP FROM CASCADE....DEG C.
C     TO = OUTSIDE HEATING MEDIUM TEMPERATURE.
C     DTO = TIME DERIVATIVE OF HEATING MEDIUM TEMPERATURE...DEG C/S.
C     AI(J) = FOURIER COSINE SERIES COEFFICIENTS OF THE
C             HEATING TEMPERATURE TRAJECTORIES.
C     OMEGA(J) = FREQUENCY IN THE COSINE SERIES....RAD/S.
C     FEED = STEADY STATE FLUID FLOW THROUGH CASCADE...KG/HR.
C     WF = STEADY STATE FLUID FLOW THROUGH CASCADE...KG/SEC.
C     CPF = FLUID HEAT CAPACITY...J/KG-DEG C.
C     CPM = METAL WALL   "          "
C     AKW = METAL WALL THERMAL CONDUCTIVITY .....WATTS/M-DEG K.
C     RHOF = DENSITY OF PROCESS FLUID.... KG/CU M.
C     RHO = DENSITY OF METAL WALL.....KG/CU M.
C     XL = THICKNESS OF THE HEAT TRANSFER WALL ..... M.
C     QINIT = INITIAL HEAT TRANSFER RATE ACROSS THE WALL.....DEG. C.
C     DTDX = INITIAL TEMP. GRADIENT IN WALL .... DEG.C/M.
C     DTFILM = INITIAL TEMP. AT WALL INSIDE SURFACE..... DEG. C.
C     TIWALL = INITIAL TEMP. AT THE INNER WALL SURFACE.... DEG. C.
C     TWINIT = INITIAL VALUE OF THE (INTERNAL) WALL TEMP.....DEG. C.
C     DTINIT = INITIAL VALUE OF THE  "      "    DERIVATIVE ... DEG/M.
C     TXL(N) = VECTOR OF INSIDE (PROCESS) WALL TEMPERATURES .... DEG. C.
C     ZL = LENGTH OF HEAT TRANSFER SURFACE....M.
C     DIA = INSIDE DIAMETER OF HEAT TRANSFER SHELL.... M.
C     CIRC = CIRCUMFERENCE OF H. T. SHELL.... M.
C     SPEED = ROTOR SPEED M/S.
```

```
C     TPOWER = TOTAL ROTOR POWER INPUT.....WATTS.
C     POWERN = POWER INPUT FOR 1 STAGE  " .
C
C
C
C
C.....................................................................
C
C
C                    DVERK VARIABLE DEFINITIONS:
C
C     L = NO. EQUATIONS (DEPENDENT VARIABLES) TO BE INTEGRATED.
C     LW = ROW DIMENSION OF WORK MATRIX W (GENERALLY = L).
C     W = WORK MATRIX GENERALLY OF DIMENSION (LW,9).
C     IER = ERROR PARAMETER...SEE IMSL DOC.
C     IND = INDICATOR.  CAN BE INPUT OR OUTPUT
C           ON INPUT: IND = 1 CAUSES DEFAULT OPTIONS TO BE USED.
C                     IND = 2 REQUIRES SETTING FIRST 9 ELEMENTS
C                     OF COMMUNICATIONS VECTOR 'C'.
C           ON OUTPUT: IND = 2 OR 3 USUALLY INDICATES TOL TOO SMALL.
C
C     TOL = CONVERGENCE TOLERANCE ON RK TIME INTEGRATION.
C     C = COMMUNICATIONS VECTOR.  USED TO SELECT OPTIONS AND TO
C         RETAIN INFORMATION BETWEEN CALLS....SEE IMSL DOC.
C
C
C.....................................................................
C
C
C
C                    DPDES VARIABLE DEFINITIONS:
C
C        NPDES - NO. OF (FIRST ORDER IN TIME) INPUT P.D.E.S
C        FCN - EXTERNAL SUBROUTINE TO EVALUATE P.D.E.S
C        BNDRY -   "         "      "      "    BOUNDARY COND.
C        TIME2 - INDEPENDENT VARIABLE
C        H2 - INPUT=FIRST STEP SIZE IN TIME2;  OUTPUT=LAST STEP SIZE
C        TEND2 - VALUE OF TIME2 WHERE SOLUTION DESIRED
C        NX - NO. OF MESH POINTS
C        X(I) - VECTOR OF (NX) BREAK POINTS IN THE SPACE DESCRITIZATION
C        Y(NPDES,2,NX) - HOLDS INITIAL COND. ON INPUT
C                        AND SOLUTION ON OUTPUT:
C                        Y(K,1,I) = U(K) AT X(I)
C                        Y(K,2,I) = UX(K) AT X(I)
C        IY - FIRST DIMENSION OF Y
C        TOL2 - ERROR BOUND ON TIME DESCRITIZATION
C        INDEX - I/O PARAMETER.....SEE IMSL DOCUMENTS
C        WK - WORK VECTOR OF LENGTH NR*NPDES*(30*NPDES+36) +5*NPDES
C             + 2*NR
C        IER - ERROR PARAMETER (OUTPUT) SEE IMSL DOCUMENTS
C
C
C
C
C**********************************************************************
C
C
C                           MAIN PROGRAM
C
C
C**********************************************************************
C
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL FLUX
      DIMENSION TEMP(30), DTEMP(30), C(24), W(50,9), WK(1000),
```

```
     &          TIME(30), TEND(30), TOL2(30), X(10),
C
     &          HO(30), HI(30), AREA(30), FMASS(30), WMASS(30),
     &          TXL(30), TNOW(30), TLAST(30), AI(40), OMEGA(40),
     &          QAI(40), QOMEGA(40),
C
     &          Y1(1,2,10), Y2(1,2,10), Y3(1,2,10), Y4(1,2,10),
     &          Y5(1,2,10), Y6(1,2,10), Y7(1,2,10), Y8(1,2,10),
     &          Y9(1,2,10), Y10(1,2,10), Y11(1,2,10), Y12(1,2,10),
     &          Y13(1,2,10), Y14(1,2,10), Y15(1,2,10), Y16(1,2,10),
     &          Y17(1,2,10), Y18(1,2,10), Y19(1,2,10), Y20(1,2,10),
     &          Y21(1,2,10), Y22(1,2,10), Y23(1,2,10), Y24(1,2,10),
     &          Y25(1,2,10), Y26(1,2,10), Y27(1,2,10), Y28(1,2,10),
     &          Y29(1,2,10), Y30(1,2,10)
C
      COMMON AREA, HO, HI, FMASS, TNOW, TLAST, TXL, TO, TOLAST,
     &       TEMPIN, WF, CPF, CPM, XL, AKW, RHO, DTIME, AI, OMEGA,
     &       DTO, QINIT, NSTAGE
C
      EXTERNAL FUNC, FCN, BNDRY
C
      DATA QAI /121.2880D0, -3.18535D0, -2.31093D0,
     &    -3.67265D0, -3.81757D0, -2.60488D0,
     &    -1.92128D0, -1.79375D0, -1.43949D0, -1.01605D0, -0.74390D0,
     &    -0.66791D0, -0.62152D0, -0.45197D0, -0.42515D0, -0.43820D0,
     &    -0.40316D0, -0.36566D0, -0.36730D0, -0.37851D0, -0.29154D0,
     &    -0.29771D0, -0.32882D0, -0.26816D0, -0.25490D0, -0.24333D0,
     &    -0.27232D0, -0.24363D0, -0.19714D0, -0.20570D0, -0.19409D0,
     &    -0.14801D0, -0.17173D0, -0.11414D0, -0.17927D0, -0.14045D0,
     &    -0.12832D0, -0.13221D0, -0.11819D0, -0.10831D0/
C
      DATA QOMEGA /0.D0, 0.049087D0, 0.098174D0,
     &     0.147262D0, 0.196349D0, 0.245436D0,
     &     0.294524D0, 0.343611D0, 0.392698D0, 0.441786D0, 0.490873D0,
     &     0.539960D0, 0.589048D0, 0.638135D0, 0.687222D0, 0.736310D0,
     &     0.785397D0, 0.834484D0, 0.883572D0, 0.932659D0, 0.981746D0,
     &     1.030834D0, 1.079921D0, 1.129008D0, 1.178096D0, 1.227183D0,
     &     1.276270D0, 1.325358D0, 1.374445D0, 1.423532D0, 1.472620D0,
     &     1.521707D0, 1.570795D0, 1.619882D0, 1.668969D0, 1.718057D0,
     &     1.767144D0, 1.816231D0, 1.865319D0, 1.914406D0/
C
C.....................................................................
C
C
C  LOAD THE ARRAYS AI AND OMEGA WITH THEIR CONSTANT VALUES....THIS IS
C  NECESSARY BECAUSE DATA STATEMENT CANNOT BE USED WITH VARIABLE
C  STORED IN COMMON BLOCK.
C
      DO 5 J=1,40
         AI(J) = QAI(J)
         OMEGA(J) = QOMEGA(J)
 5    CONTINUE
C
C
C  INPUT HEAT TRANSFER COEFFICIENTS
C
      AVHO = 1455.D0
      AVHI = 829.0D0
C
C  INPUT INITIAL VALUES OF TEMPERATURES
C
      TEMPIN = 14.83D0
      TMPOUT = 19.23D0
      TO = 90.7D0
      DTO = 0.D0
C
```

```
C  INPUT UNIT PARAMETERS
C
      N = 24
      FEED = 118.D0
      SPEED = 5.03D0
      HOLDUP = .0630D0
C
C  INPUT PROCESS CONSTANTS
C
      CPF = 4187.D0
      CPM = 501.D0
      RHOF = 996.D0
      RHO = 8023.D0
      AKW = 16.3D0
      XL = .005D0
      ZL = .455D0
      PI = 3.14159D0
      DIA = .08D0
      CIRM = PI * DIA
      WF = FEED / 3600.D0
C
C  SET THE NUMBER OF TANKS IN SERIES FOR THE MODEL.
C
      AN = DFLOAT(N)
C
C  INITIALIZE 'TOTALIZING' VARIABLES.
C
      TAREA = 0.D0
      TMWALL = 0.D0
C
C  INITIALIZE PROCESS ARRAYS AND SUM THE TOTALIZED VARIABLES.
C
      DO 1 J=1,N
         TIME(J) = 0.D0
         HO(J) = AVHO
         HI(J) = AVHI
         AREA(J) = CIRM * ZL/AN
         TAREA = TAREA + AREA(J)
         FMASS(J) = HOLDUP / AN
         WMASS(J) = RHO * XL * CIRM * ZL/AN
         TMWALL = TMWALL + WMASS(J)
 1    CONTINUE
C
C                 ..............................
C
C  THE FOLLOWING CODE BLOCK ASSIGNS INITIAL VALUES TO THE INPUT
C  PARAMETERS FOR THE EXTERNAL IMSL SUBROUTINES DVERK AND DPDES.
C
C  INITIALIZE DVERK VARIABLES.
C
      L = N
      LW = L
      IER = 0
C
C  USE DVERK DEFAULT OPTIONS.
C
      IND = 1
      TOL = 1.D-4
C
C  LOAD INITIAL PROCESS FLUID TEMPS INTO VECTOR (TEMP(L)) AND LOAD
C  INITIAL TIME DERIVATIVES OF FLUID TEMP INTO DTEMP(J) FOR USE BY
C  DVERK.  LOAD INITIAL VALUES OF THE LAST MEASURED PROCESS TEMP INTO
C  TLAST(J) AND THE CURRENT PROCESS TEMPERATURE INTO TNOW(J) FOR USE BY
C  DPDES.
C
C  NOTE INITIAL PROCESS FLUID TEMP FOR EACH STAGE (J) IS CALCULATED BY
C  LINEAR INTERPOLATION BETWEEN UNIT INLET AND OUTLET TEMPERATURES AT
```

```
C  TIME = 0.
C
C
      DO 2 J=1,L
         TEMP(J) = TEMPIN + (TMPOUT - TEMPIN) * DBLE(J) / AN
C        TEMP(J) = TEMPIN
         DTEMP(J) = 0.D0
         TXL(J) = TEMP(J)
         TNOW(J) = TEMP(J)
         TLAST(J) = TNOW(J)
 2    CONTINUE
C
C  INITIALIZE DPDES VARIABLES:
C
      NPDES = 1
      NX = 10
      IY = 1
C
      DO 3 J=1,N
         TOL2(J) = 1.D-4
 3    CONTINUE
C
C
C.....................................................................
C
C
C          ..................FOR TIME < 0  ......................
C  CALCULATE THE STEADY STATE HEAT TRANSFER RATE, THE INSIDE
C  WALL FILM TEMPERATURE DROP, AND THE WALL TEMPERATURE GRADIENT
C  PRIOR TO THE STEP CHANGE IN HEATING TEMPERATURE.
C
C
C
      QINIT = WF*CPF * (TMPOUT - TEMPIN)
      DTFILM = QINIT / (AVHI * TAREA)
      DTDX = QINIT /(AKW * TAREA)
C
C
C.....................................................................
C
C
C  THE FOLLOWING CODE LOOPS ASSIGN MESH POINTS (J) THROUGH THE
C  HEATING WALL.  AFTER THE MESH POINTS ARE ESTABLISHED THE INITIAL
C  TEMPERATURES AND TEMPERATURE GRADIENTS ARE ASSIGNED AT EACH POINT
C  IN A GIVEN STAGE (N).  THE OUTER LOOP REPEATS UNTIL ALL STAGES HAVE
C  THEIR INITIAL CONDITIONS ASSIGNED.
C
C
      DO 60 NSTAGE=1,N
C
C
C  SET DISTRIBUTION EXPONENT ON SPACE DESCRITIZATION GRID.
C
        XMESH = 1.D0
C
C  INSTALL THE INITIAL CONDITIONS ON A POWER MESH ACROSS THE WALL FOR
C  EACH STAGE ELEMENT.  NOTE THE MESH POINTS ARE CONCENTRATED AT THE
C  HEATING WALL WHEN THE VALUE OF "MESH" > 1; AND ARE EQUALLY SPACED
C  FOR "MESH" = 1.
C
        XN = DFLOAT(NX-1)
        DO 10 J=1,NX
           XJ = DFLOAT(J-1)
           X(J) = (XJ/XN)** XMESH * XL
C
C  LOAD INITIAL VALUES OF WALL TEMPERATURES AND DERIVATIVES FOR DPDES.
C  THE LOGICAL FLAG "FLUX" DIRECTS PROCESSING OF INITIAL CONDITIONS FOR
```

```
C  EITHER CONSTANT WALL TEMPERATURE OR CONSTANT WALL HEAT FLUX.
C
            FLUX = .FALSE.
C
            IF (FLUX) THEN
C
C   CALCULATE THE STARTING INSIDE WALL TEMPERATURE
C
               TIWALL = TEMP(NSTAGE) + DTFILM
               TWINIT = TIWALL + DTDX * X(J)
               DTINIT = 0.D0
            ELSEIF (X(J) .EQ. 0.D0) THEN
               TWINIT = TO
               DTINIT = 0.D0
            ELSE
               TWINIT = TEMP(NSTAGE)
               DTINIT = 0.D0
            ENDIF
C
C
            GO TO ( 201, 202, 203, 204, 205, 206, 207, 208, 209, 2010,
     &              2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
     &              2019, 2020, 2021, 2022, 2023, 2024, 2025), NSTAGE
C
C
 201        Y1(1,1,J) = TWINIT
            Y1(1,2,J) = DTINIT
            GO TO 10
C
 202        Y2(1,1,J) = TWINIT
            Y2(1,2,J) = DTINIT
            GO TO 10
C

 203        Y3(1,1,J) = TWINIT
            Y1(1,2,J) = DTINIT
            GO TO 10
C
 204        Y4(1,1,J) = TWINIT
            Y4(1,2,J) = DTINIT
            GO TO 10
C

 205        Y5(1,1,J) = TWINIT
            Y5(1,2,J) = DTINIT
            GO TO 10
C
 206        Y6(1,1,J) = TWINIT
            Y6(1,2,J) = DTINIT
            GO TO 10
C

 207        Y7(1,1,J) = TWINIT
            Y7(1,2,J) = DTINIT
            GO TO 10
C
 208        Y8(1,1,J) = TWINIT
            Y8(1,2,J) = DTINIT
            GO TO 10
C

 209        Y9(1,1,J) = TWINIT
            Y9(1,2,J) = DTINIT
            GO TO 10
C
 2010       Y10(1,1,J) = TWINIT
            Y10(1,2,J) = DTINIT
```

```
            GO TO 10
C
 2011       Y11(1,1,J) = TWINIT
            Y11(1,2,J) = DTINIT
            GO TO 10
C
 2012       Y12(1,1,J) = TWINIT
            Y12(1,2,J) = DTINIT
            GO TO 10
C
 2013       Y13(1,1,J) = TWINIT
            Y13(1,2,J) = DTINIT
            GO TO 10
C
 2014       Y14(1,1,J) = TWINIT
            Y14(1,2,J) = DTINIT
            GO TO 10
C
 2015       Y15(1,1,J) = TWINIT
            Y15(1,2,J) = DTINIT
            GO TO 10
C
 2016       Y16(1,1,J) = TWINIT
            Y16(1,2,J) = DTINIT
            GO TO 10
C
 2017       Y17(1,1,J) = TWINIT
            Y17(1,2,J) = DTINIT
            GO TO 10
C
 2018       Y18(1,1,J) = TWINIT
            Y18(1,2,J) = DTINIT
            GO TO 10
C
 2019       Y19(1,1,J) = TWINIT
            Y19(1,2,J) = DTINIT
            GO TO 10
C
 2020       Y20(1,1,J) = TWINIT
            Y20(1,2,J) = DTINIT
            GO TO 10
C
 2021       Y21(1,1,J) = TWINIT
            Y21(1,2,J) = DTINIT
            GO TO 10
C
 2022       Y22(1,1,J) = TWINIT
            Y22(1,2,J) = DTINIT
            GO TO 10
C
 2023       Y23(1,1,J) = TWINIT
            Y23(1,2,J) = DTINIT
            GO TO 10
C
 2024       Y24(1,1,J) = TWINIT
            Y24(1,2,J) = DTINIT
            GO TO 10
C
 2025       Y25(1,1,J) = TWINIT
```

```
            Y25(1,2,J) = DTINIT
            GO TO 10
C
C
 10      CONTINUE
C
 60   CONTINUE
C
C             ..............................
C
C  THE FOLLOWING CODE FORMATS THE OUTPUT HEADER DATA.
C
       WRITE (6,210) FEED, SPEED
       WRITE (6,200) N, XMESH, TAREA, HOLDUP, TMWALL, WF, TEMPIN,
     &               TMPOUT, QINIT, AVHO, AVHI
       WRITE (6,215)
C      WRITE (6,400) (I, FMASS(I), HO(I), HI(I), I=1,N)
C      WRITE (6,250) N, TAREA, HOLDUP, TMWALL, WF, TEMPIN, TO
C      WRITE (6,450) (I, FMASS(I), HO(I), HI(I), I=1,N)
C      WRITE (6,300)
C      WRITE (6,350)
C
C.....................................................................
C
C  SET THE TIME INTEGRATION INCREMENT DTIME IN SEC., AND THE INITIAL
C  STARTING TIME.
C
       DTIME = 0.05D0
       FTIME = 0.D0
C
C  THE OUTER LOOP CODE SEGMENT CARRIES OUT THE TIME INTEGRATION
C  IN STEPS.
C
       DO 20 J=1,601
          AKTIME = DBLE (J-1) * DTIME
C
C  OUTPUT THE SIMULATION RESULTS AT 1 SECOND INTERVALS
C
          REMAIN = DMOD (AKTIME, 1.D0)
          IF (REMAIN .LT. 1.D-3 .OR. REMAIN .GT. .999D0) THEN
             WRITE (6,150) AKTIME, TO, TEMP(N)
C            WRITE (6,100) KTIME, TEMP(1),TEMP(2),TXL(1), TXL(2)
C            WRITE (6,150) KTIME, (TEMP(I),I=1,N,5)
C            WRITE (6,155) (TXL(I),I=1,N,4)
C            WRITE (6,700) (TEMP(I),TNOW(I), TLAST(I),I=1,L)
          ENDIF
C
C  THE OUTER LOOP SOLVES THE 'TANKS IN SERIES' MODEL USING THE I.V.P.
C  SOLVER DVERK.
C
          FTEND = DBLE(J) * DTIME
C
C  THE INNER LOOP CALCULATES THE WALL TEMPERATURES & GRADIENTS
C  AT EACH STAGE USING THE B.V.P. SOLVER DPDES.  THE INSIDE WALL TEMPS
C  ARE PASSED TO THE COMMON BLOCK VIA VECTOR TXL FOR USE BY DVERK.
C
C
          DO 30 NSTAGE=1,N
C
             GO TO ( 301, 302, 303, 304, 305, 306, 307, 308, 309, 3010,
     &               3011, 3012, 3013, 3014, 3015, 3016, 3017, 3018,
     &               3019, 3020, 3021, 3022, 3023, 3024, 3025), NSTAGE
C
 301         TEND(1) = FTEND
             H2 = 1.D-10
             INDEX = 1
C
```

```
            CALL DPDES (NPDES, FCN, BNDRY, TIME(1), H2, TEND(1),
     &                  X, Y1, IY, NX, TOL2(1), INDEX, WK, IER2)
            TXL(1) = Y1(1,1,NX)
C           WRITE (6,150)  Y1(1,1,1), (Y1(1,1,I), I=2,NX,2)
            GO TO 30
C
 302        TEND(2) = FTEND
            H2 = 1.D-10
            INDEX = 1
C
            CALL DPDES (NPDES, FCN, BNDRY, TIME(2), H2, TEND(2),
     &                  X, Y2, IY, NX, TOL2(2), INDEX, WK, IER2)
            TXL(2) = Y2(1,1,NX)
C           WRITE (6,150) Y2(1,1,1), (Y2(1,1,I), I=2,NX,2)
            GO TO 30
C
 303        TEND(3) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(3), H2, TEND(3),
     &                  X, Y3, IY, NX, TOL2(3), INDEX, WK, IER2)
            TXL(3) = Y3(1,1,NX)
C           WRITE (6,150) Y3(1,1,1), (Y3(1,1,I), I=2,NX,2)
            GO TO 30
C
 304        TEND(4) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(4), H2, TEND(4),
     &                  X, Y4, IY, NX, TOL2(4), INDEX, WK, IER2)
            TXL(4) = Y4(1,1,NX)
C           WRITE (6,150) Y4(1,1,1), (Y4(1,1,I), I=2,NX,2)
            GO TO 30
C
 305        TEND(5) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(5), H2, TEND(5),
     &                  X, Y5, IY, NX, TOL2(5), INDEX, WK, IER2)
            TXL(5) = Y5(1,1,NX)
C           WRITE (6,150) Y5(1,1,1), (Y5(1,1,I), I=2,NX,2)
            GO TO 30
C
 306        TEND(6) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(6), H2, TEND(6),
     &                  X, Y6, IY, NX, TOL2(6), INDEX, WK, IER2)
            TXL(6) = Y6(1,1,NX)
C           WRITE (6,150) Y6(1,1,1), (Y6(1,1,I), I=2,NX,2)
            GO TO 30
C
 307        TEND(7) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(7), H2, TEND(7),
     &                  X, Y7, IY, NX, TOL2(7), INDEX, WK, IER2)
            TXL(7) = Y7(1,1,NX)
C           WRITE (6,150) Y7(1,1,1), (Y7(1,1,I), I=2,NX,2)
            GO TO 30
C
 308        TEND(8) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(8), H2, TEND(8),
     &                  X, Y8, IY, NX, TOL2(8), INDEX, WK, IER2)
```

```
            TXL(8) = Y8(1,1,NX)
C           WRITE (6,150) Y8(1,1,1), (Y8(1,1,I), I=2,NX,2)
            GO TO 30
C
 309        TEND(9) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(9), H2, TEND(9),
     &                  X, Y9, IY, NX, TOL2(9), INDEX, WK, IER2)
            TXL(9) = Y9(1,1,NX)
C           WRITE (6,150) Y9(1,1,1), (Y9(1,1,I), I=2,NX,2)
            GO TO 30
C
 3010        TEND(10) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(10), H2, TEND(10),
     &                  X, Y10, IY, NX, TOL2(10), INDEX, WK, IER2)
            TXL(10) = Y10(1,1,NX)
C           WRITE (6,150) Y10(1,1,1), (Y10(1,1,I), I=2,NX,2)
            GO TO 30
C
 3011        TEND(11) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(11), H2, TEND(11),
     &                  X, Y11, IY, NX, TOL2(11), INDEX, WK, IER2)
            TXL(11) = Y11(1,1,NX)
C           WRITE (6,150) Y11(1,1,1), (Y11(1,1,I), I=2,NX,2)
            GO TO 30
C
 3012        TEND(12) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(12), H2, TEND(12),
     &                  X, Y12, IY, NX, TOL2(12), INDEX, WK, IER2)
            TXL(12) = Y12(1,1,NX)
C           WRITE (6,150) Y12(1,1,1), (Y12(1,1,I), I=2,NX,2)
            GO TO 30
C
 3013        TEND(13) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(13), H2, TEND(13),
     &                  X, Y13, IY, NX, TOL2(13), INDEX, WK, IER2)
            TXL(13) = Y13(1,1,NX)
C           WRITE (6,150) Y13(1,1,1), (Y13(1,1,I), I=2,NX,2)
            GO TO 30
C
 3014        TEND(14) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(14), H2, TEND(14),
     &                  X, Y14, IY, NX, TOL2(14), INDEX, WK, IER2)
            TXL(14) = Y14(1,1,NX)
C           WRITE (6,150) Y14(1,1,1), (Y14(1,1,I), I=2,NX,2)
            GO TO 30
C
 3015        TEND(15) = FTEND
            H2 = 1.D-10
            INDEX = 1
            CALL DPDES (NPDES, FCN, BNDRY, TIME(15), H2, TEND(15),
     &                  X, Y15, IY, NX, TOL2(15), INDEX, WK, IER2)
            TXL(15) = Y15(1,1,NX)
C           WRITE (6,150) Y15(1,1,1), (Y15(1,1,I), I=2,NX,2)
            GO TO 30
C
```

```
 3016         TEND(16) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(16), H2, TEND(16),
     &                    X, Y16, IY, NX, TOL2(16), INDEX, WK, IER2)
              TXL(16) = Y16(1,1,NX)
C             WRITE (6,150) Y16(1,1,1), (Y16(1,1,I), I=2,NX,2)
              GO TO 30
C
 3017         TEND(17) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(17), H2, TEND(17),
     &                    X, Y17, IY, NX, TOL2(17), INDEX, WK, IER2)
              TXL(17) = Y17(1,1,NX)
C             WRITE (6,150) Y17(1,1,1), (Y17(1,1,I), I=2,NX,2)
              GO TO 30
C
 3018         TEND(18) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(18), H2, TEND(18),
     &                    X, Y18, IY, NX, TOL2(18), INDEX, WK, IER2)
              TXL(18) = Y18(1,1,NX)
C             WRITE (6,150) Y18(1,1,1), (Y18(1,1,I), I=2,NX,2)
              GO TO 30
C
 3019         TEND(19) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(19), H2, TEND(19),
     &                    X, Y19, IY, NX, TOL2(19), INDEX, WK, IER2)
              TXL(19) = Y19(1,1,NX)
C             WRITE (6,150) Y19(1,1,1), (Y19(1,1,I), I=2,NX,2)
              GO TO 30
C
 3020         TEND(20) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(20), H2, TEND(20),
     &                    X, Y20, IY, NX, TOL2(20), INDEX, WK, IER2)
              TXL(20) = Y20(1,1,NX)
C             WRITE (6,150) Y20(1,1,1), (Y20(1,1,I), I=2,NX,2)
              GO TO 30
C
 3021         TEND(21) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(21), H2, TEND(21),
     &                    X, Y21, IY, NX, TOL2(21), INDEX, WK, IER2)
              TXL(21) = Y21(1,1,NX)
C             WRITE (6,150) Y21(1,1,1), (Y21(1,1,I), I=2,NX,2)
              GO TO 30
C
 3022         TEND(22) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(22), H2, TEND(22),
     &                    X, Y22, IY, NX, TOL2(22), INDEX, WK, IER2)
              TXL(22) = Y22(1,1,NX)
C             WRITE (6,150) Y22(1,1,1), (Y22(1,1,I), I=2,NX,2)
              GO TO 30
C
 3023         TEND(23) = FTEND
              H2 = 1.D-10
              INDEX = 1
              CALL DPDES (NPDES, FCN, BNDRY, TIME(23), H2, TEND(23),
```

```
     &                    X, Y23, IY, NX, TOL2(23), INDEX, WK, IER2)
           TXL(23) = Y23(1,1,NX)
C          WRITE (6,150) Y23(1,1,1), (Y23(1,1,I), I=2,NX,2)
           GO TO 30
C
 3024      TEND(24) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(24), H2, TEND(24),
     &                    X, Y24, IY, NX, TOL2(24), INDEX, WK, IER2)
           TXL(24) = Y24(1,1,NX)
C          WRITE (6,150) Y24(1,1,1), (Y24(1,1,I), I=2,NX,2)
           GO TO 30
C
 3025      TEND(25) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(25), H2, TEND(25),
     &                    X, Y25, IY, NX, TOL2(25), INDEX, WK, IER2)
           TXL(25) = Y25(1,1,NX)
C          WRITE (6,150) Y25(1,1,1), (Y25(1,1,I), I=2,NX,2)
           GO TO 30
C
 3026       TEND(26) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(26), H2, TEND(26),
     &                    X, Y26, IY, NX, TOL2(26), INDEX, WK, IER2)
           TXL(26) = Y26(1,1,NX)
C          WRITE (6,150) Y26(1,1,1), (Y26(1,1,I), I=2,NX,2)
           GO TO 30
C
 3027       TEND(27) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(27), H2, TEND(27),
     &                    X, Y27, IY, NX, TOL2(27), INDEX, WK, IER2)
           TXL(27) = Y27(1,1,NX)
C          WRITE (6,150) Y27(1,1,1), (Y27(1,1,I), I=2,NX,2)
           GO TO 30
C
 3028       TEND(28) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(28), H2, TEND(28),
     &                    X, Y28, IY, NX, TOL2(28), INDEX, WK, IER2)
           TXL(28) = Y28(1,1,NX)
C          WRITE (6,150) Y28(1,1,1), (Y28(1,1,I), I=2,NX,2)
           GO TO 30
C
 3029       TEND(29) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(29), H2, TEND(29),
     &                    X, Y29, IY, NX, TOL2(29), INDEX, WK, IER2)
           TXL(29) = Y29(1,1,NX)
C          WRITE (6,150) Y29(1,1,1), (Y29(1,1,I), I=2,NX,2)
           GO TO 30
C
 3030       TEND(30) = FTEND
           H2 = 1.D-10
           INDEX = 1
           CALL DPDES (NPDES, FCN, BNDRY, TIME(30), H2, TEND(30),
     &                    X, Y30, IY, NX, TOL2(30), INDEX, WK, IER2)
           TXL(30) = Y30(1,1,NX)
C          WRITE (6,150) Y30(1,1,1), (Y30(1,1,I), I=2,NX,2)
```

```
            GO TO 30
C
 30       CONTINUE
C
C                  ....................................
C
C
C  UPDATE THE VECTOR OF PREVIOUS PROCESS TEMPS THAT WILL BE PASSED TO
C  DPDES SUBROUTINE BNDRY.
C
         DO 40 NSTAGE=1,N
            TLAST(NSTAGE) = TEMP(NSTAGE)
 40      CONTINUE
C
C  NOW SOLVE FOR THE FLUID TEMPERATURES AT THE END OF THE TIME
C  INCREMENT IN EACH OF THE N STAGES BY INTEGRATING THE  "TANKS IN
C  SERIES" MODEL USING THE I.V.P. SOLVING ROUTINE DVERK.
C
C
         CALL DVERK (L, FUNC, FTIME, TEMP, FTEND, TOL, IND, C, LW, W,
     &                IER)
C
C
C  UPDATE VECTOR OF CURRENT PROCESS TEMPERATURES TO BE PASSED TO
C  DPDES SUBROUTINE BNDRY.
C
         DO 50 NSTAGE=1,N
            TNOW(NSTAGE) = TEMP(NSTAGE)
 50      CONTINUE
C
 20    CONTINUE
       STOP
C
C.....................................................................
C
C  INPUT / OUTPUT FORMAT STATEMENTS
C
C
 200   FORMAT (/,T20,'DISTRIBUTED WALL MODEL ',/
     &         /,T20,'NO. OF STAGE ELEMENTS = ',I2,
     &         /,T20,'MESH DISTRIBUTION EXPONENT = ',F6.4,
     &         /,T20,'TOTAL HEAT TRANS. AREA SQ.M. = ',F6.4,
     &         /,T20,'LIQUID HOLDUP IN UNIT KG = ',F6.4,
     &         /,T20,'TOTAL THERMAL WALL MASS KG = ',F6.2,
     &         /,T20,'LIQUID FEED RATE KG/S = ',F6.4,
     &         /,T20,'LIQUID FEED TEMP. DEG. C = ',F6.2,
     &         /,T20,'INITIAL LIQ. DISCH. TEMP. DEG. C = ',F6.2,
     &         /,T20,'ROTOR POWER INPUT WATTS = ',F6.1,
     &         /,T20,'HEATING FILM H. T. COEF. WATTS/SQ.M = ',F7.1,
     &         /,T20,'LIQUID FILM H. T. COEF. WATTS/SQ.M = ',F6.1,///)
C
 215   FORMAT (///,T20,'  TIME',T28,'  TO',T36,'  TEMP',/)
C
 210   FORMAT ('1',/////,T20,'FEED RATE KG/HR = ',F5.1,2X,
     &          'ROTOR SPEED M/S = ',F4.2,/)
C
 250   FORMAT ('1',T45,'DISTRIBUTED WALL MODEL - ENG. UNITS',/
     &         /,T45,'NO. OF STAGE ELEMENTS = ',I2,
     &         /,T45,'TOTAL HEAT TRANS. AREA = ',F6.2,
     &         /,T45,'LIQUID HOLDUP IN UNIT = ',F6.2,
     &         /,T45,'TOTAL THERMAL WALL MASS = ',F6.2,
     &         /,T45,'LIQUID FEED RATE = ',F6.2,
     &         /,T45,'LIQUID FEED TEMP. = ',F6.2,
     &         /,T45,'HEATING TEMP. = ',F6.2,///)
C
 400   FORMAT (10X,'STAGE: ',I2,3X,'HOLDUP = ',F5.4,3X,
     &         'HEATING FILM COEF. = ',F7.2,3X,'PROCESS FILM COEF. = ',
```

```
     &         F6.2)
C
 450  FORMAT (10X,'STAGE: ',I2,3X,'HOLDUP = ',F5.2,3X,
     &         'HEATING FILM COEF. = ',F7.2,3X,'PROCESS FILM COEF. = ',
     &         F6.2)
C
 100  FORMAT (1X,T10,I4,T20,F6.2,T30,F6.2,T40,F6.2,T50,F6.2)
C
 150  FORMAT (20X,7(F6.2,2X))
C
 155  FORMAT ('+',T70,7(F6.2,2X))
C
 300  FORMAT (///T10,'TIME SEC.',T25,'TEMP. IN',T40,'WALL INS. TEMP.',
     &         T55,'DISCH. TEMP.',/)
C
 350  FORMAT (///8X,'TIME',T15,'T(1)',4X,'T(6)',4X,'T(11)',4X,'T(16)',
     &         4X,'T( )',T71,'TW(1)',3X,'TW(6)',3X,'TW(11)',
     &         3X,'TW(16)',3X,'TW( )',/)
C
 600  FORMAT (1X,7(E10.3,1X))
C
 700  FORMAT (1X,5(E14.7,1X))
C
 710  FORMAT (F14.7)
C
      END
C
C
C*********************************************************************************
C
C
C                          SUBROUTINE FUNC
C
C*********************************************************************************
C
C
      SUBROUTINE FUNC (L, TIME, TEMP, DTEMP)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TEMP(30), TXL(30), DTEMP(30), T(30), TM(30),
     &          HO(30), HI(30), AREA(30), FMASS(30), TNOW(30),
     &          TLAST(30), AI(40), OMEGA(40)
C
      COMMON AREA, HO, HI, FMASS, TNOW, TLAST, TXL, TO, TOLAST,
     &       TEMPIN, WF, CPF, CPM, XL, AKW, RHO, DTIME, AI, OMEGA,
     &       DTO, QINIT, NSTAGE
C
C
C  REDEFINE TEMP VECTORS AS FLUID AND METAL WALL COMPONENTS FOR EACH
C  STAGE N.
C
      DO 20 N=1,L
         T(N) = TEMP(N)
         TM(N) = TXL(N)
 20   CONTINUE
C
C  CALCULATE POWER INPUT FORM ROTOR FOR A SINGLE STAGE.
C  IN THIS CASE TOTAL POWER IS SET TO QINIT, THE AMOUNT OF
C  SENSIBLE HEATING OF THE PROCESS FLUID AT TIME < 0.
C  THE ASSUMPTION IS THAT ALL ENERGY INPUT TO THE FLUID PRIOR
C  TO THE HEATING TEMP. STEP INPUT IS DUE TO THE ROTOR.
C  TO USE THIS MODEL, THE FLAG .FLUX. IN THE MAIN PGM. MUST BE
C  SET TO "FALSE".
C
      TPOWER = QINIT
      POWERN = TPOWER / DBLE(L)
C
```

```
C  EVALUATE THE ENERGY BALANCE EQUATIONS [1] AT EACH STAGE N.
C  FOR STAGE NO. 1 USE CONSTANT INLET TEMPERATURE TEMPIN.
C
      DO 10 N=1,L
         IF (N .EQ. 1) THEN
            DTEMP(N) = ( WF*CPF*(TEMPIN-T(N)) + AREA(N)*HI(N)*
     &                   (TM(N)-T(N)) + POWERN ) / (FMASS(N)*CPF)
C
         ELSE
            DTEMP(N) = ( WF*CPF*(T(N-1)-T(N)) + AREA(N)*HI(N)*
     &                   (TM(N)-T(N)) + POWERN ) / (FMASS(N)*CPF)
C
         ENDIF
 10   CONTINUE
      RETURN
      END
C
C
C*********************************************************************
C
C
C                          SUBROUTINE FCN
C
C
C*********************************************************************
C
C
C
      SUBROUTINE FCN (NPDES, X, TIME, U, UX, UXX, UTIME)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(NPDES), UX(NPDES), UXX(NPDES), UTIME(NPDES),
     &          AREA(30), HO(30), HI(30), FMASS(30), TNOW(30),
     &          TLAST(30),TXL(30), AI(40), OMEGA(40)
C
      COMMON AREA, HO, HI, FMASS, TNOW, TLAST, TXL, TO, TOLAST,
     &       TEMPIN, WF, CPF, CPM, XL, AKW, RHO, DTIME, AI, OMEGA,
     &       DTO, QINIT, NSTAGE
C
C
C
C  CALCULATE THERMAL DIFFUSIVITY (ALPHA) AND SUBSTITUTE INTO THE
C  DIFFUSION EQUATION [3].
C
         ALPHA = (RHO * CPM) / AKW
         UTIME(1) = UXX(1) / ALPHA
C        WRITE (6,100) ALPHA, RHO, CPM, AKW, UTIME(1), UXX(1)
 100     FORMAT (1X,3(D10.3))
  10  RETURN
      END
C
C
C
C*********************************************************************
C
C
C                       SUBROUTINE BNDRY
C
C
C*********************************************************************
C
C
C
      SUBROUTINE BNDRY (NPDES, XI, TIME, ALP, BET, GAMP)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ALP(2), BET(2), GAMP(2), TEMP(30),
```

```
     &          AREA(30), HO(30), HI(30), FMASS(30), TNOW(30),
     &          TLAST(30),TXL(30), AI(40), OMEGA(40)
C
      COMMON AREA, HO, HI, FMASS, TNOW, TLAST, TXL, TO, TOLAST,
     &       TEMPIN, WF, CPF, CPM, XL, AKW, RHO, DTIME, AI, OMEGA,
     &       DTO, QINIT, NSTAGE
C
C
C
C  FOR THE OUTSIDE WALL BOUNDARY CONDITIONS AT X=0;
C  THE HEATING TEMP/TIME DERIVATIVE OF EQUATIONS [4] IS EXPRESSED AS
C  THE DERIVATIVE OF A FOURIER COSINE SERIES REPRESENTING THE HEATING
C  TEMPERATURE AS A FUNCTION OF TIME.
C
      IF (XI .EQ. 0.D0) THEN
         ALP(1) = -HO(NSTAGE)
         BET(1) = AKW
         CALL TOPRIM (TIME)
         GAMP(1) = -HO(NSTAGE) * DTO
C
C  INSIDE WALL BOUNDARY CONDITIONS AT X=L; NOTE (TEMP-TLAST)/DTIME
C  IS TREATED AS THE INSIDE WALL PROCESS TEMP/TIME DERIVATIVE OF
C  EQUATIONS [4].
C
      ELSE
         ALP(1) = HI(NSTAGE)
         BET(1) = AKW
         GAMP(1) = HI(NSTAGE) * (TNOW(NSTAGE) - TLAST(NSTAGE)) /
     &             DTIME
C     WRITE (6,100) TNOW(NSTAGE), TLAST(NSTAGE), GAMP(1)
      ENDIF
 100  FORMAT (1X,3(D10.2,1X))
      RETURN
      END
C
C
C*********************************************************************************
C
C
C                          SUBROUTINE TOPRIM
C
C
C*********************************************************************************
C
C
      SUBROUTINE TOPRIM (TIME)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AREA(30), HO(30), HI(30), FMASS(30), TNOW(30),
     &          TLAST(30),TXL(30), AI(40), OMEGA(40)
C
      COMMON AREA, HO, HI, FMASS, TNOW, TLAST, TXL, TO, TOLAST,
     &       TEMPIN, WF, CPF, CPM, XL, AKW, RHO, DTIME, AI, OMEGA,
     &       DTO, QINIT, NSTAGE
C
C
C  INITIALIZE THE FOURIER SERIES SUMMATIONS TO ZERO.
C
      TO = 0.D0
      DTO = 0.D0
C
C  SELECT THE NO. OF TERMS TO BE EVALUATED IN THE SERIES.  AT SMALL
C  TIMES MORE TERMS ARE REQUIRED TO OBTAIN A FAITHFUL REPLICATION OF
C  THE TEMPERATURE FUNCTION
C
      IF (TIME .GE. 7.D0) THEN
         J = 10
```

```
      ELSEIF (TIME .GE. 4.D0) THEN
         J = 20
      ELSE
         J = 40
      ENDIF
C
C  EVALUATE THE SERIES FOR HEATING TEMPERATURE AND ITS TIME DERIVATIVE.
C  NOTE: AI'S ARE THE PREDETERMINED FOURIER COEFFICIENTS AND OMEGA'S
C  ARE THEIR CORRESPONDING FREQUENCIES. BOTH ARE PASSED TO TOPRIM
C  THROUGH THE COMMON BLOCK.
C
      DO 10 I=1,J
         OMEGAT = OMEGA(I) * TIME
         TO = TO + AI(I) * DCOS (OMEGAT)
         DTO = DTO - AI(I) * DSIN (OMEGAT) * OMEGA(I)
 10   CONTINUE
C
C     WRITE (6,100) TIME, TO, DTO, J
 100  FORMAT (1X,3(F10.4,2X),I3)
C
      RETURN
      END
```

```
//A020KIMH JOB 33649,HUNTER,MSGLEVEL=(2,0),REGION=1024K
// EXEC SAS
//SYSIN DD *
TITLE 'COEFFICIENTS FOR A QUADRATIC HEAT TRANSFER SPACE MODEL';
*************************************************************;
*                                                           ;
*          THIS INPUTS THE DATA                             ;
*                                                           ;
*************************************************************;
DATA ONE;
INPUT HOUT HIN SSE I $;
HOUT2 = HOUT*HOUT; HIN2 = HIN*HIN;
HINOUT = HIN*HOUT;
*************************************************************;
*                                                           ;
*          THE ACTUAL DATA                                  ;
*                                                           ;
*************************************************************;
CARDS;
11000 50 8302 a
11000 600 8814 C
 9500 300 315  E
 8000 150 2237 F
 8000 300 243 K
 8000 450 2868 L
 6500 300 162 m
 5000 50  8626 n
 5000 600 5462 o
*************************************************************;
*                                                           ;
*          THIS SECTION COMPUTES THE REGRESSION             ;
*          COEFFICIENTS, PLOTS THE DESIGN SPACE,            ;
*          AND PRINTS THE DATA                              ;
*                                                           ;
*************************************************************;
PROC REG;
MODEL SSE = HOUT HIN HINOUT HOUT2 HIN2 / INFLUENCE;
PROC PLOT; PLOT HOUT*HIN=I;
PROC PRINT;
```

```
//C2035GGV JOB 316B5,VINING,MSGLEVEL=(2,0),REGION=1024K
// EXEC SAS
//SYSIN DD *
TITLE;
PROC MATRIX PRINT;
BO = -4.39539;
BI = -24.64852;
BIO = .0024775;
BOO = .00025936;
BII = .215407;
A = 2*BOO; C = 2*BII;
B = (BIO//C)||(A//BIO);
Z = BO//BI;
H = INV(B)*(-Z);
;
*           THE ACTUAL DATA                                          ;
*                                                                    ;
*********************************************************************;
CARDS;
11000 50 8302 a
11000 600 8814 C
 9500 300 315  E
 8000 150 2237 F
 8000 300 243 K
 8000 450 2868 L
 6500 300 162 m
 5000 50  8626 n
 5000 600 5462 o
*********************************************************************;
*                                                                    ;
*           THIS SECTION COMPUTES THE REGRESSION                     ;
*           COEFFICIENTS, PLOTS THE DESIGN SPACE,                    ;
*           AND PRINTS THE DATA                                      ;
*                                                                    ;
*********************************************************************;
PROC REG;
MODEL SSE = HOUT HIN HINOUT HOUT2 HIN2 / INFLUENCE;
PROC PLOT; PLOT HOUT*HIN=I;
PROC PRINT;
```

VITA

Kim R Hunter was born in Detroit on October 21, 1949. He received an S.B. degree from the chemical engineering department of the Massachusetts Inst. of Technology.

Kim Hunter

Kim R Hunter